

**Volume 86, Number 7,
July 2025**

**ISSN 0005-1179
CODEN: AURCAT**



AUTOMATION AND REMOTE CONTROL

**Editor-in-Chief
Andrey A. Galyaev**

<http://ait.mtas.ru>

Automation and Remote Control

Vol. 86, No. 7, July 2025

Available via license: CC BY 4.0

Automation and Remote Control

ISSN 0005-1179

Editor-in-Chief

Andrey A. Galyaev

Deputy Editors-in-Chief M.V. Khlebnikov and E.Ya. Rubinovich

Coordinating Editor A.S. Samokhin

Editorial Board

F.T. Aleskerov, A.V. Arutyunov, N.N. Bakhtadze, A.A. Bobtsov, P.Yu. Chebotarev, A.G. Chkhartshvili, L.Yu. Filimonyuk, A.L. Fradkov, O.N. Granichin, M.F. Karavai, E.M. Khorov, M.M. Khrustalev, A.I. Kibzun, S.A. Krasnova, A.P. Krishchenko, A.G. Kushner, N.V. Kuznetsov, A.A. Lazarev, A.I. Lyakhov, A.I. Matasov, S.M. Meerkov (USA), R.V. Mescheryakov, A.I. Mikhali'skii, B.M. Miller, O.V. Morzhin, R.A. Munasypov, A.V. Nazin, A.S. Nemirovskii (USA), D.A. Novikov, A.Ya. Oleinikov, P.V. Pakshin, D.E. Pal'chunov, A.E. Polyakov (France), V.Yu. Protasov, L.B. Rapoport, I.V. Rodionov, N.I. Selvesyuk, P.S. Shcherbakov, A.N. Sobolevski, O.A. Stepanov, A.B. Tsybakov (France), D.V. Vinogradov, V.M. Vishnevskii, K.V. Vorontsov, and L.Yu. Zhilyakova

Staff Editor E.A. Martekhina

SCOPE

Automation and Remote Control is one of the first journals on control theory. The scope of the journal is control theory problems and applications. The journal publishes reviews, original articles, and short communications (deterministic, stochastic, adaptive, and robust formulations) and its applications (computer control, components and instruments, process control, social and economy control, etc.).

Automation and Remote Control is abstracted and/or indexed in *ACM Digital Library*, *BFI List*, *CLOCKSS*, *CNKI*, *CNPIEC Current Contents/Engineering, Computing and Technology*, *DBLP*, *Dimensions*, *EBSCO Academic Search*, *EBSCO Advanced Placement Source*, *EBSCO Applied Science & Technology Source*, *EBSCO Computer Science Index*, *EBSCO Computers & Applied Sciences Complete*, *EBSCO Discovery Service*, *EBSCO Engineering Source*, *EBSCO STM Source*, *EI Compendex*, *Google Scholar*, *INSPEC*, *Japanese Science and Technology Agency (JST)*, *Journal Citation Reports/Science Edition*, *Mathematical Reviews*, *Naver*, *OCLC WorldCat Discovery Service*, *Portico*, *ProQuest Advanced Technologies & Aerospace Database*, *ProQuest-ExLibris Primo*, *ProQuest-ExLibris Summon*, *SCImago*, *SCOPUS*, *Science Citation Index*, *Science Citation Index Expanded (Sci-Search)*, *TD Net Discovery Service*, *UGC-CARE List (India)*, *WTI Frankfurt eG*, *zbMATH*.

Journal website: <http://ait.mtas.ru>

© The Author(s), 2025 published by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences.

Automation and Remote Control participates in the Copyright Clearance Center (CCC) Transactional Reporting Service.

Available via license: CC BY 4.0

0005-1179/25. *Automation and Remote Control* (ISSN: 0005-1179 print version, ISSN: 1608-3032 electronic version) is published monthly by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow 117997, Russia.

Volume 86 (12 issues) is published in 2025.

Publisher: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences.

65 Profsoyuznaya street, Moscow 117997, Russia; e-mail: redacsia@ipu.rssi.ru; <http://ait.mtas.ru>, <http://ait-arc.ru>

Contents

Automation and Remote Control

Vol. 86, No. 7, 2025

Control in Technical Systems

- A Hybrid Method for Integrating Separated Sensors for the Adaptive Identification of Stationary and Mobile Objects under Prior Uncertainty
Yu. G. Bulychev 599
- Program Control of a Multichannel Object Search System with a Finite Number of Processing Lines in Each Channel
V. V. Khutortsev 613
- Developing a Multi-Output Soft Sensor Based on an Error Predictor for Quality Control of the Target Product of a Complex Distillation Column
D. V. Shtakin, A. A. Plotnikov, O. Yu. Snegirev, and A. Yu. Torgashov 628
-

Control in Social Economic Systems

- A Control Algorithm for Player's Actions in a Triopoly Game under Linear Demand and Cost Functions
M. I. Geraskin 644
-

Intellectual Control Systems, Data Analysis

- A Modification of the Multidimensional Fuzzy Controller with the Local Term Division of Input Variables
A. F. Antipin, E. V. Antipina, and S. A. Mustafina 655
- Randomized Machine Learning Methods for Generating Random Data Ensembles with Given Numerical Characteristics
Yu. S. Popkov, A. Yu. Popkov, and Yu. A. Dubnov 666
-

Optimization, System Analysis, and Operations Research

- On the Emergence of Binary Mapping Systems
A. N. Karkishchenko 683
-
-

A Hybrid Method for Integrating Separated Sensors for the Adaptive Identification of Stationary and Mobile Objects under Prior Uncertainty

Yu. G. Bulychev

JSC Concern Radioelectronic Technologies, Moscow, Russia

e-mail: profbulychev@yandex.ru

July 14, 2024

February 25, 2025

February 27, 2025

Abstract—A novel adaptive identification method for stationary and moving objects is developed within the combined application of the classical statistical approach based on probabilistic models and an unconventional approach oriented to the principles of continuity, multiplication, and clustering for a set of arbitrary spatially distributed sensors. The problem is solved under essential prior uncertainty associated, e.g., with the presence of weakly formalized abnormal measurement errors, faults and complete failures of individual sensors, or significant changes in the observation conditions of an identified object. Identification models, criterion, and algorithm robust to such uncertainty are presented. As a comparative analysis example, a triangulation system of sensors is used, and the effect achieved through adaptation is shown for this system. Practical recommendations are given.

Keywords: essential prior uncertainty, sensor, information and measurement system, identified object, integration, hybrid method, adaptation, abnormal measurement errors, continuity principle, multiplication principle, clustering principle, objective function, family of partial estimates, competing cluster, identification algorithm, posterior weights, resulting estimate

DOI: 10.31857/S0005117925070012

1. INTRODUCTION

By now, many theoretical and practical advancements have been made for building various information and measurement systems (IMSS) consisting of spatially distributed sensors (homogeneous, inhomogeneous, stationary, and mobile), used for jointly identifying the state of stationary and moving objects. Such systems are widespread in different fields: medicine, fault diagnosis, radio astronomy, active and passive radar, electronic intelligence, field tests, security, etc.

Among the IMSS, there is an entire class of systems characterized by the following operating conditions: multistage data processing, real-time mode, essential uncertainty, and high risks (damage) from poor-quality assessments and decisions made on their basis. Uncertainty may be associated with unreliable knowledge of the distribution laws of measurement errors, skips in observations, the occurrence of equipment faults and complete failures, the presence of abnormal measurement errors (AMEs), etc.; for example, see [1–5]. Concerning possible risks, by assumption, it is important to consider not only the possible average damage but also the losses from a poor-quality solution in each particular case, i.e., when handling a fixed sample of measurements, particularly of a small size. Such samples are encountered at the stages of secondary and tertiary data processing, e.g., in active and passive radar systems (when forming the indirect measurements of bearings, time delays, slant ranges, radial velocities, etc.). The quality of such measurements determines the

efficiency of estimating the location and motion parameters of observed objects within the well-known coordinate measurement methods (triangulation, difference ranging, triangulation difference ranging, etc.).

Data processing algorithms under uncertainty are based on the classical probabilistic methods of maximum posterior probability density or maximum likelihood, usually in an extended version [1, 6–8]. In this version, all unknown uncertainty parameters are included in the expanded state vector, but in practice, the expansion procedure sharply increases computational cost and leads to the well-known effect of “smearing accuracy” and problems with the convergence and stability of the estimates. The above difficulties of classical methods are even further aggravated: within the multi-alternative approach [6], uncertainty is often eliminated by introducing a family of different hypotheses, each to be either rejected or confirmed. Analysis shows that for uncertainty conditions and strict requirements for IMSs, classical probabilistic methods are often of little use, but they can be effectively used at the mathematical modeling stage to study the potential capabilities of the IMSs being designed.

There exist other optimal and quasi-optimal estimation methods that can be used under uncertainty. They include the extended least squares (XLS) method [7], the minimum method of geometric and kinematic residuals [9], the method of invariants [10], various methods of linear and nonlinear filtering (e.g., Kalman) [1, 6, 7, 11], different adaptive methods with parameter tuning [12, 13], robust methods [6, 14, 15], randomized methods [16, 17], cluster methods [17, 18], and neural network methods [19], as well as numerous heuristics used in special-purpose IMSs (for example, see [20–25]). When imposing several constraints on observation conditions, these methods allow overcoming uncertainty factors (e.g., AMEs). However, under essential uncertainty, these methods encounter various unforeseen limitations and strict requirements for IMSs, which significantly narrow the scope of their application.

This is often due to the difficulty of considering the geometric factor, which depends on the spatial arrangement of the sensor family and the observed object. For example, triangulation-incorrect observation domains arise for goniometric systems, which are typical even in the case of good measurements. In addition, spatial estimation problems often involve pronounced nonlinearity, with the use of unbounded (e.g., some trigonometric functions, necessary for describing the relationship between the estimated and measured parameters), etc.

As a rule, the above methods guarantee convergence on average (either over an ensemble of realizations or on a single sufficiently long realization), not ensuring a high-quality estimate based on one small sample. In addition, the estimation procedures mainly use weight processing, in which weights are calculated based on a large volume of prior information. Under the specified uncertainty, such information may often be absent or turn out to be unreliable.

The general trend to build IMSs capable of withstanding different uncertainty factors is to integrate various estimation methods considering the limits of their applicability (including those mentioned above). Each method of a selected family should be “accuracy-orthogonal,” in some sense, to other methods of this family and realize its potential capabilities for definite observation conditions. A hybrid method implementing the principle of integration should combine the advantages of both the classical statistical approach (ensuring optimality on average) and the unconventional approach (ensuring optimality in partial), covering to the maximum extent the specifics of an applied IMS and the uncertainty conditions.

For the unconventional approach, one can use the cluster-variant method [18], which has proven to be an effective tool for eliminating prior uncertainty, both on average and in partial. It involves the principle of multiplication of single estimates (considering a set of solution variations for the same problem with different sets of measurements based on an initial input observation), the clustering principle (dividing these estimates into several competing clusters), and an algorithm for

constructing the resulting estimate based on these clusters. This method suffers from the following drawbacks. First, only a triangulation system of goniometric (i.e., homogeneous) sensors is considered. Second, the identified object is supposed to be stationary (only its location is estimated). Third, the resulting estimate neglects the different accuracies of single estimates within one cluster. (This disagrees with the important principle of adaptability of the identification algorithm for a wider range of observation conditions.)

Below, we develop a generalized cluster-variant identification method (GCVIM) to completely remove all these disadvantages of the cluster-variant method [18]. The method should be based on principles understandable to experts in the field of construction and application of this class of IMSs in order to form guaranteed estimates (within the accepted constraints) of the location and, moreover, motion parameters of identified objects under essential uncertainty. The weights for measurement processing should take into account both prior and posterior information.

2. BASIC DEFINITIONS, MODELS, AND CONSTRAINTS. PROBLEM STATEMENT

By analogy with [18], to omit cumbersome notation and considerations, the method proposed is described mainly for the case of a single object. The method will be generalized to the case of several objects in Section 5.

In the Cartesian frame XYZ , we consider a system consisting of a set of sensors D_m , $m = \overline{1, M}$, where the position of each sensor is given by a vector $\mathbf{p}_m = [\rho_{xm}, \rho_{ym}, \rho_{zm}]^T$. An object is characterized by the vector of estimated constant parameters $\boldsymbol{\lambda} = [\lambda_p, p = \overline{1, P}]^T$, and all measured parameters are combined into the vector $\mathbf{s} = [s_j, j = \overline{1, J}]^T$. For a moving object, the components of the vector $\boldsymbol{\lambda}$ represent the unknown spectral coefficients at the corresponding basis functions of a selected functional space. For a stationary object, $\boldsymbol{\lambda}$ includes the Cartesian coordinates of the object. Depending on the type of sensors used, the components of the vector $\mathbf{s}$ can be, e.g., the values of the following quantities: azimuth and elevation angle, slant range, radial velocity, as well as the level (amplitude or power), phase difference, or time delays of the received signal (including various derivatives of these quantities), related to different points in space and time instants.

The input observation is characterized by a vector

$$\mathbf{h} = \mathbf{s} + \boldsymbol{\Delta} + \boldsymbol{\xi}, \quad (1)$$

where $\boldsymbol{\Delta} = [\Delta_j, j = \overline{1, J}]^T$ is an unknown measurement error and $\boldsymbol{\xi} = [\xi_j, j = \overline{1, J}]^T$ is measurement noise with an unknown distribution law but zero mean and a given correlation matrix $\mathbf{K}_\xi$.

To describe the error $\boldsymbol{\Delta}$, we use the following approach. Let J^Δ correspond to the real number of those nonzero components of the vector $\boldsymbol{\Delta}$ that are AMEs and lead to unreliable identification results. Assume that $J^\Delta \leq J_{\max}^\Delta$, where $J_{\max}^\Delta \in \{0, 1, \dots\}$, and

$$J_{\min}^{\mathbf{h}} + J_{\max}^\Delta \leq J, \quad (2)$$

where $J_{\min}^{\mathbf{h}}$ is the minimum number of components of the vector $\mathbf{h}$ sufficient for the high-quality identification of the object, given no AMEs.

Below, various sets of measurements will be formed on the basis of $\mathbf{h} : \mathbf{h}_i = [h_{ij}, j = \overline{1, J_i}]^T$, where $h_{ij} \in \{h_j, j = \overline{1, J}\}$, $J_i \leq J$, $\mathbf{h}_i \neq \mathbf{h}_r \ \forall i \neq r, i, r = \overline{1, I}$. For these sets, we introduce the constraint

$$J_{\min}^{\mathbf{h}} \leq J_i \leq J. \quad (3)$$

With this descriptive approach to the error Δ , some of its nonzero components can take arbitrary values (including anomalous), they can be located in the vector Δ in the most arbitrary way, and there is no universal and satisfactory model to describe them. The only way to formalize the error Δ is to introduce some quantitative constraints (like (2) and (3)), which correspond to the operation practice of a particular IMS.

The parameter vector λ is subject to the traditional constraints

$$\lambda \in \Lambda \subset R^P, \quad (4)$$

where $\Lambda = \{\Lambda_{\min}^p \leq \lambda_p \leq \Lambda_{\max}^p\}$, $\Lambda_{\min}^p \in R^1$, and $\Lambda_{\max}^p \in R^1$.

By assumption, we know an operator Ψ such that

$$\Psi : \mathbf{s} \rightarrow \lambda, \quad (5)$$

i.e., $\lambda = \Psi(\mathbf{s})$; in addition, there exists an inverse operator

$$\Psi^{-1} : \lambda \rightarrow \mathbf{s} \quad (6)$$

for which $\mathbf{s} = \Psi^{-1}(\lambda)$.

GCVIM involves three major principles: continuity, multiplication, and clustering. According to the first principle, the quality of identification continuously depends on the parameters of the input observation of a sufficient dimension. In particular, a sequential decrease in the parameter J (by excluding certain components of the vector $\mathbf{h}$) smoothly improves the accuracy of the resulting estimate. The second principle states that it is possible to form a set of partial estimates (both good and bad) and, due to conditions (2)–(4), the set of good estimates has sufficient cardinality. By the third principle, under conditions (2)–(4), all possible partial estimates can be clustered, and among the competing clusters there exists an optimal one, a basis for forming a reliable resulting estimate.

According to the multiplication principle, we associate the main vector $\mathbf{h}$ with a family of partial vectors $\{\mathbf{h}_i, i = \overline{1, I}\}$:

$$\mathbf{h}_i = \mathbf{s}_i + \Delta_i + \xi_i, \quad (7)$$

where $\mathbf{s}_i = [s_{ij}, j = \overline{1, J_i}]^T$, $\Delta_i = [\Delta_{ij}, j = \overline{1, J_i}]^T$, and $\xi_i = [\xi_{ij}, j = \overline{1, J_i}]^T$.

The choice of I and $J_1, \dots, J_I$ should agree with conditions (2) and (3); moreover, among $\{\mathbf{h}_i, i = \overline{1, I}\}$ there should exist vectors without AMEs.

Each vector $\mathbf{h}_i$ can be assigned a certain operator Ψ^* of classical optimal estimation (e.g., the maximum likelihood operator):

$$\Psi^* : (\mathbf{h}_i, \mathbf{K}_{\xi_i}) \rightarrow \lambda_i^*, \quad (8)$$

where $\lambda_i^* = \Psi^*(\mathbf{h}_i, \mathbf{K}_{\xi_i})$ is a partial estimate characterized by the correlation matrix $\mathbf{K}_{\xi_i}^*$ of estimation errors and $\mathbf{K}_{\xi_i}$ is the correlation matrix of the noise ξ_i .

The elements of the matrix $\mathbf{K}_{\xi_i}^*$ are calculated using well-known formulas (for example, see [1, 3, 4]); they are completely determined by the IMS type (triangulation, difference ranging, angle-power, etc.), its geometry and observation conditions, as well as the matrix $\mathbf{K}_{\xi_i}$.

For each estimate λ_i^* , the vector of secondary measurements $\left(\mathbf{s}_i^* = [s_{ij}^*, j = \overline{1, J_i}]^T\right)$ is estimated using the known operator Θ (which also depends on the type, geometry, and parameters of the IMS):

$$\Theta : \lambda_i^* \rightarrow \mathbf{s}_i^*, \quad (9)$$

where $\mathbf{s}_i^* = \Theta(\lambda_i^*)$.

Note that generally, the dimension J_i of the vector $\mathbf{s}_i^*$ does not coincide with that (J) of the vector $\mathbf{s}$. The vector $\mathbf{s}_i^*$ is calculated by considering the Cartesian coordinates of only those IMS sensors that are used to form the partial estimate vector $\mathbf{h}_i$.

Secondary measurements allow discarding bad partial estimates λ_i^* by checking the condition

$$|s_{ij}^* - h_{ij}| \leq \varepsilon_{ij} \quad \forall j = \overline{1, J_i}, \quad (10)$$

where ε_{ij} is the maximum value of the residual calculated considering the IMS type and the matrix $\mathbf{K}_i^*$.

If condition (10) fails or the estimate λ_i^* does not agree within the constraint (4), this partial estimate will be eliminated. Let $\lambda_{[n]}^*$, $n = \overline{1, N}$ denote the partial estimates remaining after elimination, where $\lambda_{[n]}^* \in \{\lambda_1^*, \dots, \lambda_I^*\}$. Obviously, the set of such estimates is non-empty under conditions (2) and (3).

Definition 1. A scalar error Δ_{ij} , representing one component of the vector Δ_i , is said to be called abnormal if it violates condition (10). Otherwise, the error Δ_{ij} is said to be normal.

Definition 2. Any group of scalar errors (two or more) representing components of the vector Δ_i is said to be anomalous (even if each of these components is itself normal) if it violates condition (10). Otherwise, the group is said to be normal.

Definition 3. The “clogging” coefficient of an original sample $\mathbf{h}$ is the value $k_{\mathbf{h}} = 100(k_{\Delta}/J)[\%]$, where k_{Δ} denotes the total number of abnormal scalar errors and scalar errors included in abnormal groups.

The generated set $\mathbf{X}^* = \{\lambda_{[1]}^*, \dots, \lambda_{[N]}^*\}$ is divided into competing clusters K_q , $q = \overline{1, Q}$. (Any of the well-known clustering algorithms can be used, e.g., [18, 26, 27].) The estimates $\lambda_{[1]}^*, \dots, \lambda_{[N]}^*$ are sequentially (step by step) combined into groups: first the closest, and then those with increasing distance to each other. At the first step, each estimate $\lambda_{[n]}^*$ is treated as a separate cluster.

The clustering algorithm sets a mapping of the form $\Upsilon: \mathbf{X}^* \rightarrow \{K_q, q = \overline{1, Q}\}$, and the following condition should hold:

$$\mathbf{X}^* = \bigcup_{q=1}^Q K_q, \quad (11)$$

where $K_k \cap K_r = \emptyset$, $k, r \in \{1, \dots, Q\}$, $k \neq r$, and $K_q \neq \emptyset \quad \forall q = \overline{1, Q}$.

For each cluster, we will use the representation $K_q = \{\lambda_{[q,n]}^*, n = \overline{1, N_q}\}$, where $\lambda_{[q,n]}^*$ is a partial estimate that falls into the cluster K_q and is assigned the number n . By analogy with (9), an estimate of the vector of secondary full-dimensional measurements $\mathbf{s}_{[q,n]}^* = [s_{j[q,n]}^*, j = \overline{1, J}]^T$ is associated with each $\lambda_{[q,n]}^*$:

$$\mathbf{s}_{[q,n]}^* = \Theta(\lambda_{[q,n]}^*). \quad (12)$$

By full-dimensionality we mean that the dimensions of the vectors $\mathbf{s}$ and $\mathbf{s}_{[q,n]}^*$ coincide.

The choice of the family of clusters $\{K_1, \dots, K_Q\}$ was discussed in detail in [18]; in particular, a new adaptive clustering algorithm was proposed therein, including the case of unknown Q . This algorithm divides the entire family $\{\lambda_{[1]}^*, \dots, \lambda_{[N]}^*\}$ into clusters autonomously (without operator intervention), is simple enough for computer implementation, and requires reasonable computational cost. According to the application results, this algorithm has high effectiveness and good asymptotic properties.

Based on (1)–(12), it is required to develop an adaptive GCVIM, in an optimal formulation, that has robustness to the error Δ and constructs the resulting estimate $\lambda^* = [\lambda_p^*, p = \overline{1, P}]^T$ considering the different accuracies of partial estimates included in competing clusters. The method should include a justification of the objective function, criterion, and algorithm for selecting an optimal cluster and constructing an optimal resulting estimate.

3. JUSTIFICATION AND CONSTRUCTION OF THE OBJECTIVE FUNCTION. CRITERION AND IDENTIFICATION ALGORITHM

In the case where $\Delta = \mathbf{0}$ and the only prior information available about the noise ξ is the correlation matrix $\mathbf{K}_\xi$, preference is given to the traditional statistical method, which uses the weight matrix $\mathbf{W}_\xi = \mathbf{K}_\xi^{-1}$. (This ensures optimization on average.) Within GCVIM, under essential uncertainty, it is necessary to form a new objective function that would incorporate all the parameters figuring in the constraints (2)–(4) and characterizing competing clusters. This function should include weights that, unlike any optimal statistical method, are calculated a posteriori based on one fixed sample of measurements. (This ensures optimization in partial.)

To solve the identification problem, we use the following dimensionless normalized objective function:

$$F(\lambda, q) = J^{-1} \sum_{j=1}^J \left\{ \left[\frac{s_j(\lambda) - h_j}{\delta_j} \right]^2 w(h_j, q) \right\}, \quad 0 \leq F(\lambda, q) \leq 1, \quad (13)$$

where $s_j(\lambda)$ is a known dependence of the measured parameter s_j on λ ; $w(h_j, q)$ is the posterior weight of the measurement h_j , $0 \leq w(h_j, q) \leq 1$; $\delta_j > 0$ is a coefficient limiting the value $|s_j(\lambda) - h_j|$ for all viewed (model) values of $\lambda \in \Lambda$ and providing the corresponding normalization.

To find the weights, by analogy with (10), it is necessary to check the condition

$$|s_{j[q,n]}^* - h_j| \leq \varepsilon_{j[q,n]} \quad (14)$$

for the cluster K_q and any of partial estimates $\lambda_{[q,n]}^*$ in this cluster.

The left-hand side in (14) characterizes the residual between the primary h_j and secondary $s_{j[q,n]}^*$ measurements for all possible values of j , q , and n . To consider the constraint on this residual, we use the indicator function

$$\varphi(p) = \begin{cases} 1 - p, & 0 \leq p \leq 1, \\ \varphi(p) = 0, & p > 1. \end{cases} \quad (15)$$

With $p = (s_{j[q,n]}^* - h_j)^2 / \varepsilon_{j[q,n]}^2$, the weights covering all the accepted constraints of the identification problem and the results of multiplying the partial estimates and their clustering, are determined as

$$w(h_j, q) = N_q^{-1} \sum_{n=1}^{N_q} \varphi \left(\frac{(s_{j[q,n]}^* - h_j)^2}{\varepsilon_{j[q,n]}^2} \right), \quad 0 \leq w(h_j, q) \leq 1, \quad (16)$$

where N_q^{-1} acts as a normalizing factor.

Formula (16) takes into account the number N_q of partial estimates in each cluster K_q , the thresholds $\varepsilon_{j[q,n]}$ of admissible errors in secondary measurements, and the presence of gross measurement errors (single and group) by introducing the indicator function $\varphi(p)$. The coefficient

$w(h_j, q)$ shows the specific contribution of the measurement h_j to the formation of all marks of the cluster K_q . It can be called the coefficient of matching between the measurement h_j and the elements of the cluster K_q . The higher value this coefficient takes, the more confidence there will be in the corresponding primary measurement h_j .

In view of (14)–(16), the desired objective function takes the form

$$F(\boldsymbol{\lambda}, q) = (JN_q)^{-1} \sum_{j=1}^J \sum_{n=1}^{N_q} \left[\frac{s_j(\boldsymbol{\lambda}) - h_j}{\delta_j} \right]^2 \varphi \left(\frac{(s_{j[q,n]}^* - h_j)^2}{\varepsilon_{j[q,n]}^2} \right). \quad (17)$$

The optimal identification criterion becomes

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}, q^*), \quad (18)$$

$$q^* = \arg \max_q w(\mathbf{h}, q) = \arg \max_q \left\{ J^{-1} \sum_{j=1}^J w(h_j, q) \right\}, \quad 0 \leq w(\mathbf{h}, q) \leq 1, \quad (19)$$

where $w(\mathbf{h}, q)$ is an integral dimensionless normalized weight.

Minimizing the decision function $F(\boldsymbol{\lambda}, q)$ with respect to the vector argument $\boldsymbol{\lambda}$ leads to the equation

$$\left[\frac{\partial F(\boldsymbol{\lambda}, q)}{\partial \boldsymbol{\lambda}} \right]^T = \mathbf{0}. \quad (20)$$

By expanding the partial derivatives in (20), we obtain the system of scalar equations

$$\sum_{j=1}^J \left\{ \left(\frac{s_j(\boldsymbol{\lambda}) - h_j}{\delta_j} \right) w(h_j, q) \frac{\partial s_j(\boldsymbol{\lambda})}{\partial \lambda_p} \right\} = 0, \quad p = \overline{1, P}, \quad (21)$$

where partial derivatives are expanded given the known dependence $\mathbf{s} = \boldsymbol{\Psi}^{-1}(\boldsymbol{\lambda})$.

System (21) yields the partial estimates $\boldsymbol{\lambda}^*(q)$, $q = \overline{1, Q}$. As the resulting estimate $\boldsymbol{\lambda}^* \in \{\boldsymbol{\lambda}^*(1), \dots, \boldsymbol{\lambda}^*(Q)\}$, we choose the partial estimate $\boldsymbol{\lambda}^*(q^*)$, $q^* \in \{1, \dots, Q\}$, that satisfies, according to (19), the following criterion for a fixed $\mathbf{h}$:

$$w(\mathbf{h}, q^*) > w(\mathbf{h}, q) \quad \forall q = \overline{1, Q}, \quad q \neq q^*. \quad (22)$$

By choosing appropriate values of the parameters in conditions (2)–(4) and (10) and introducing appropriate ranges for the values of I , Q , and $\mathbf{K}_\xi$, one can always ensure a unique solution of the identification problem with a required accuracy. With the correct planning of the measurement experiment (for example, see [7]), the resulting cluster K_{q^*} with the largest number of partial estimates formed by the reliable observations $\mathbf{h}_i$ is almost always detected. In turn, the parameters $\varepsilon_{j[q,n]}$ significantly affect the accuracy of identification.

The introduction of constraints (4) allows cutting the so-called incorrect identification domains, in which the uniqueness condition of the solution may be violated. For example, consider a triangulation IMS consisting of two direction finders; for such domains, the triangle formed by the base and two lines of sight has a too small angle at its apex (the object is located in it). In these cases, the lines of sight are almost parallel, and even minor bearing errors, not to mention AMEs, can cause significant coordinate measurement errors. In addition, there may appear competing clusters that are in the same direction, are separated by large distances, and have approximately the same number of elements. Such incorrectness is tackled either by introducing constraints (4) or

by using additional measurements from some external system that, together with the IMS under consideration, eliminates incorrect domains. This also applies to optimal experimental planning problems.

If we consider only one cluster, remove the accepted constraints, and let the weights $w(h_1, q), \dots, w(h_J, q)$ be the reciprocal values of the variances of the primary measurement errors (distributed according to the Gaussian law), the solution (18)–(21) will correspond to the well-known maximum likelihood estimate for the case of no AMEs.

In the classical statistical formulation, the identification problem is solved based on all measurements (regardless of whether they are good or bad), which undergo joint optimal processing considering pre-assigned prior weights. The novel hybrid method also operates all measurements but takes into account both prior information (in the form of the weight matrices $\mathbf{W}_{\xi_i} = \mathbf{K}_{\xi_i}^{-1}$ used to construct partial estimates) and posterior information (in the form of the weights $w(h_1, q), \dots, w(h_J, q)$ and $w(\mathbf{h}, q)$), which is required to find the optimal cluster and construct the resulting estimate.

The algorithm for solving the identification problem in the cluster optimal formulation includes the following steps:

- 1) The partial estimates are divided into clusters $\mathbf{K}_q$, $q = \overline{1, Q}$ (see clustering recommendations in [18]).
- 2) For each cluster $\mathbf{K}_q$, the number of marks (N_q) in it, as well as the partial $w(h_1, q), \dots, w(h_J, q)$ and integral $w(\mathbf{h}, q)$ weights, are calculated.
- 3) The number q^* of the optimal cluster $\mathbf{K}_{q^*}$ is found using the criterion (19).
- 4) For the cluster $\mathbf{K}_{q^*}$, the system of equations (21) is solved, with the initial condition in the form of a partial estimate taken as the center of this cluster. Finally, in view of (21), we obtain the resulting estimate $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^*(q^*) = [\lambda_p(q^*), p = \overline{1, P}]^T$ of the parameter vector $\boldsymbol{\lambda}^*$ of the identified object.

Remark 1. Based on the linear approximation of the residuals $s_j(\boldsymbol{\lambda}) - h_j$, the well-known approximate approach to constructing the maximum likelihood estimate yields the corresponding system of linear algebraic equations instead of (21).

The implementation of the proposed algorithm (with the preparation of initial data) requires the computational cost $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$, where Γ_1 is the cost of constructing a family of partial estimates, Γ_2 is the cost of clustering these estimates and selecting the resulting cluster, and Γ_3 is the cost of obtaining the resulting estimate. The implementation of these operations in special computing environments causes no particular difficulties and can be done in real time.

Formulas (13)–(22) form the mathematical basis of GCVIM.

4. APPLICATION OF THE NOVEL METHOD TO A TRIANGULATION IMS

Consider a triangulation IMS with two-channel direction finders used as sensors (D_m , $m = \overline{1, M}$), to measure the azimuth α_m and elevation angle β_m of a stationary emitting object. In this case, we have $P = 3$, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T = [x, y, z]^T$, $J = 2M$, $\mathbf{s} = [\alpha_m, \beta_m, m = \overline{1, M}]^T$, $\mathbf{h} = [h_j, j = \overline{1, J}]^T = [h_m^\alpha, h_m^\beta, m = \overline{1, M}]^T$, and $\delta_j = \pi$.

As $\mathbf{s} = \mathbf{P}\mathbf{s}\mathbf{i}^{-1}(\boldsymbol{\lambda})$ the well-known formulas are applied:

$$\begin{cases} \alpha_m(\boldsymbol{\lambda}) = \arccos \left\{ (x - x_m) \left[(x - x_m)^2 + (y - y_m)^2 \right]^{-1/2} \right\}, \\ \beta_m(\boldsymbol{\lambda}) = \arcsin \left\{ (z - z_m) \left[(x - x_m)^2 + (y - y_m)^2 + (z - z_m)^2 \right]^{-1/2} \right\}. \end{cases} \quad (23)$$

The system of equations (21) is formed based on (23).

Condition (14) turns into

$$\begin{cases} |\alpha_{m[q,n]}^* - h_m^\alpha| \leq \varepsilon_{m[q,n]}^\alpha, \\ |\beta_{m[q,n]}^* - h_m^\beta| \leq \varepsilon_{m[q,n]}^\beta, \end{cases} \quad (24)$$

and the objective function becomes

$$F(\boldsymbol{\lambda}, q) = \sum_{m=1}^M \left[\left(\frac{\alpha_m(\boldsymbol{\lambda}) - h_m^\alpha}{\pi} \right)^2 w(h_m^\alpha, q) + \left(\frac{\beta_m(\boldsymbol{\lambda}) - h_m^\beta}{\pi} \right)^2 w(h_m^\beta, q) \right], \quad (25)$$

$$\begin{cases} w(h_m^\alpha, q) = N_q^{-1} \sum_{n=1}^{N_q} \varphi \left(\frac{(\alpha_{m[q,n]}^* - h_m^\alpha)^2}{(\varepsilon_{m[q,n]}^\alpha)^2} \right), \\ w(h_m^\beta, q) = N_q^{-1} \sum_{n=1}^{N_q} \varphi \left(\frac{(\beta_{m[q,n]}^* - h_m^\beta)^2}{(\varepsilon_{m[q,n]}^\beta)^2} \right). \end{cases} \quad (26)$$

The integral weight is calculated as follows:

$$w(\mathbf{h}, q) = M^{-1} \sum_{m=1}^M [w(h_m^\alpha, q) + w(h_m^\beta, q)]. \quad (27)$$

As an example, we take the planar problem by letting $\rho_{zm} = 0$ and $\lambda_3 = z = 0$, i.e., $\beta_m = 0 \ \forall m = \overline{1, M}$. Assume that the matrix of primary azimuthal errors is diagonal: $\mathbf{K}_\xi = \text{diag}[\sigma_{\alpha 1}^2, \dots, \sigma_{\alpha M}^2]$. In this case, using the known operator Ψ^* , we can form a family of partial estimates $\boldsymbol{\lambda}_i^* = [x_i^*, y_i^*]^T$, where

$$\begin{cases} x_i^* = (B_i E_i - C_i D_i)(A_i C_i - B_i^2)^{-1}, \\ y_i^* = (B_i D_i - A_i E_i)(A_i C_i - B_i^2)^{-1}. \end{cases}$$

Here, $A_i = \sum_{m=1}^{M_i} \cos^2 \alpha_{mi} / \gamma_{mi}$, $B_i = 2^{-1} \sum_{m=1}^{M_i} \sin 2\alpha_{mi} / \gamma_{mi}$, $C_i = \sum_{m=1}^{M_i} \sin^2 \alpha_{mi} / \gamma_{mi}$, $D_i = \sum_{m=1}^{M_i} \varphi_{mi} / \gamma_{mi}$, $E_i = \sum_{m=1}^{M_i} \mu_{mi} / \gamma_{mi}$, $\gamma_{mi} = \tilde{R}_{mi}^2 \sigma_{\alpha mi}^2$, $\varphi_{mi} = \rho_{xmi} \cos^2 \alpha_{mi} - 2^{-1} \rho_{ymi} \sin 2\alpha_{mi}$, $\mu_{mi} = 2^{-1} \rho_{xmi} \sin 2\alpha_{mi} - \rho_{ymi} \sin^2 \alpha_{mi}$, and $\tilde{R}_{mi}$ is the approximate range from the object to the sensor D_{mi} , $m \in \{1, \dots, M_i\}$.

For the triangulation IMS, we construct estimation error ellipses and, in view of the partial estimates and the geometry of the triangulation IMS, use these ellipses to calculate the constants $\varepsilon_{m[q,n]}^\alpha$ for each cluster and partial estimates in it.

Formulas (25)–(27) form the basis of the one-stage GCVIM for the triangulation IMS. Unlike the well-known cluster-variant triangulation estimation method [18], when forming posterior weights, the novel method considers the fact that the right-hand sides of inequalities (24) depend both on the competing cluster number q and on the partial estimate number n in this cluster.

It is also possible to use the two-stage GCVIM for the triangulation IMS in two versions. The first version is based on the possibility to initially implement the two-stage GCVIM only on azimuthal channels (the first stage) to form a family of planar partial estimates and clusters and then determine the estimates x^* and y^* of the two Cartesian coordinates of the object (x and y , respectively).

A particular line of the object's position corresponds to the point (x^*, y^*) in the three-dimensional space. Combining this line and all possible position cones corresponding to the elevation angles, one constructs the missing estimate z^* of the spatial coordinate z (the second stage).

The second version is to find, in relation only to azimuth channels, unreliable azimuth channels using the matching coefficients $w(h_m^\alpha, q)$. The measurements of such channels are excluded from further processing. Then, on the remaining (priority) azimuth channels and all elevation channels, GCVIM is implemented in full. In this case, the number of partial estimates analyzed increases (compared to the first version), which improves the quality of triangulation estimation under uncertainty.

Obviously, the two-stage GCVIM for the two versions above does not exhaust the potential capabilities of the optimal GCVIM under uncertainty but is less costly from a computational viewpoint. This is primarily due to a significant reduction in the number of partial triangulation estimates and competing clusters.

5. GENERALIZATION OF THE METHOD TO THE CASE OF MANY OBJECTS

The extension of GCVIM to this case depends on the purpose, design principles, and organization of the computing process in a particular IMS. We mention two basic versions of GCVIM implementation in the case of many objects.

Version 1. A separate stage is to divide measurements into classes based on their belonging to a given object (the measurement identification problem). Such a truncated problem formulation is quite common in practice under the corresponding decomposition of the computational process. In this case, GCVIM is applied to each class identified. For this, it is necessary to form the measurement vectors $\mathbf{h}_{(l)} = [h_{(l)j}, j = \overline{1, J_{(l)}}]^T$, where $l \in \overline{1, L}$ denotes the object (class) number. For the l th class, by analogy with Section 3, the elements $K_{(l)q}$, $M_{(l)q}$, $w_{(l)}(h_j, q)$, $w_{(l)}(\mathbf{h}, q)$, and $q_{(l)}^*$ are formed, and then the resulting estimates $\lambda_{(l)}^* = [\lambda_{(l)j}^*, j = \overline{1, J}]^T$, $l = \overline{1, L}$, are constructed on their basis. Version 1, the most efficient from a computational viewpoint, allows organizing L parallel processing channels for measurements, but it does not exhaust the potential joint processing capabilities for available measurements.

Note 1. If incorrect decisions are made when solving the identification problem, then the false measurements falling into a wrong class can be treated as measurements with gross errors. GCVIM allows one to effectively struggle with such errors regardless of their nature.

Note 2. It is possible to solve the identification problem by engaging ancillary measurements, which often "load" basic measurements. (For example, in radar and electronic intelligence, these may include the carrier frequency, pulse repetition period, pulse duration, the type of intra-pulse modulation, etc.) The joint use of basic and ancillary measurements is often necessary for deeper identification of objects.

Version 2. In this case, GCVIM is applied to all measurements at once, forming clusters K_q , $q = \overline{1, Q}$. Initially, it is necessary to determine the numbers $q_{(l)}^* \in \{1, 2, \dots, Q\}$ of priority (optimal) clusters $K_{q_{(l)}^*}$, which is achieved by introducing the criterion $w_{(l)}(\mathbf{h}, q_{(l)}^*) \geq \gamma$, where γ is a given threshold for object recognition, $\gamma > 0$. Next, for all these clusters, the resulting estimates are constructed: $\lambda_{(l)}^* = [\lambda_{(l)j}^*, j = \overline{1, J}]^T$, $l = \overline{1, L}$.

Version 2 is quite costly from a computational viewpoint, as it yields a large number Q of clusters. The advantage of this version is that the identification problem for the case of many objects realizes the potential capabilities of GCVIM under uncertainty.

6. COMPARATIVE ANALYSIS

Consider a triangulation IMS in which the sensors D_m are located on a circle: $\mathbf{p}_m = [\rho_{xm}, \rho_{ym}, \rho_{zm}]^T$, $\rho_{xm} = 10^4 \cos[2\pi(m-1)/M]$, $\rho_{ym} = 10^4 \sin[2\pi(m-1)/M]$, $\rho_{zm} = 0$, and $M = 5$, $m = \overline{1, 5}$. For a stationary object, we investigated the set of its possible positions, with each position described by an individual vector of the form $\mathbf{\lambda}_k = [x_k, y_k, z_k]^T$, $x_k = 5 \times 10^4 \cos(2\pi k/K)$, $y_k = 5 \times 10^4 \sin(2\pi k/K)$, $z_k = 3 \times 10^3$, $k = \overline{1, K}$, where $K = 180$. From this point onwards, the coordinates of the object and sensors are specified in meters whereas the azimuth, elevation angle, and bearing errors in radians. For each k , it was assumed that the measurement noises of different equally accurate sensors are independent and obey the Gaussian distribution with zero mean, and the correlation matrix of the sensor D_m has the diagonal form $\mathbf{K}_m = \text{diag}[\sigma_\alpha^2, \sigma_\beta^2]$, where $\sigma_\alpha = \pi/360$ and $\sigma_\beta/360$. Errors were formed by a random number generator. The numbers of unreliable channels (no more than half of all azimuth channels and no more than half of all elevation channels) containing AMEs were selected randomly. In the channels, the resulting measurement error was formed as the sum of the AME and the admissible random error, with the possible ranges $(3\sigma_\alpha, \pi/6)$ for the azimuth channel and $(3\sigma_\beta, \pi/6)$ for the elevation channel.

For comparative analysis, four algorithms corresponding to the well-known cluster-variant method (CVM, see [28]) and GCVIM (the novel method) were considered: A_1 is the CVM algorithm with a predetermined number Q of clusters; A_2 and A_3 are the one- and two-stage CVM algorithms, respectively, with an optimally chosen number Q of clusters for a given initial condition Q_0 ; A_4 is the GCVIM algorithm. The algorithms were compared in terms of accuracy and computational efficiency.

The estimation procedure was carried out for each k with subsequent averaging over 100 experiments. In Algorithm A_1 , $Q = 7$; in Algorithms A_2 and A_3 , $Q_0 = 2$ was set as the initial condition. Two numerical characteristics were used to compare the algorithms: $S(A_i)$, the integral accuracy characteristic expressed in meters, and $T(A_i)$, the computational efficiency characteristic expressed in seconds.

For the integral accuracy characteristic, we have

$$S(A_i) = \sum_{k=1}^{180} S_k(A_i) = \frac{2\pi}{180} \sum_{k=1}^{180} \tau_k(A_i),$$

where $\tau_k(A_i) = \|\bar{\mathbf{\lambda}}_k^*(A_i) - \mathbf{\lambda}_k\|_2$ is the partial residual corresponding to the k th position of the object,

$$\bar{\mathbf{\lambda}}_k^*(A_i) = \sum_{p=1}^{100} \mathbf{\lambda}_{kp}^*(A_i)/100$$

is the average estimate of the vector $\mathbf{\lambda}_k$ (over 100 experiments), and $\mathbf{\lambda}_{kp}^*(A_i)$ is the partial estimate for the k th position of the object in the p th experiment.

Algorithms A_i were compared in terms of the relative integral accuracy characteristic

$$\delta S(A_i) = 100S(A_i)S^{-1}(A_1)[\%]$$

and the relative computational efficiency

$$\delta T(A_i) = 100T(A_i)T^{-1}(A_4)[\%],$$

as A_1 is less accurate and A_4 requires more time compared to the other algorithms. The simulation results are presented in the table below.

Table

Algorithms (A_i)	A_1	A_2	A_3	A_4
$\delta S(A_i)$, %	100	50	67	39
$\delta T(A_i)$, %	81	92	14	100

Clearly, in the abnormal operation conditions of the triangulation IMS, the modified algorithm A_4 , which considers the posterior change in weights depending on the object's position, the competing cluster number, and the position of each partial estimate in this cluster, is significantly better (in terms of accuracy) than the alternative algorithms. According to the simulation results, by analogy with [18], under the above abnormal observation conditions, the maximum likelihood method is not applicable in either a simple or extended version. For instance, the error of coordinate measurement based on the IMS under consideration reaches 100% in several directions.

7. CONCLUSIONS

The GCVIM proposed in this paper allows extending the idea of cluster-variant estimation from the simplest problem of determining the location of a stationary object in a triangulation IMS to a more complex one of identifying the parameter vector of a moving object in an arbitrary-type IMS with homogeneous and even heterogeneous measurements related to arbitrary points in space and time. The method produces a resulting estimate of the parameter vector that is robust to the effect of gross measurement errors (e.g., abnormal single and group ones) under essential prior uncertainty. GCVIM, both independently and in combination with traditional statistical approaches, can be effectively used to improve the existing and develop promising IMSs. In the case of limited computing resources and/or a large number of sensors and a high dimension of the observation vector, the two-stage version of GCVIM implementation can significantly increase the efficiency of identification.

Modern progress in the field of parallel computing (especially based on new operation principles [29, 30]) gives hope that the principle of multiplying partial estimates and their clustering in the optimal variant will not become an obstacle to the practical implementation of GCVIM in promising real-time IMSs. However, such an approach is feasible not for all existing IMSs: on large samples, it may require too many parallel data processing channels and significant computational cost.

REFERENCES

1. Chernyak, V.S., *Mnogopozitsionnaya radiolokatsiya* (Multiposition Radiolocation), Moscow: Radio i Svyaz', 1993.
2. Kupriyanov, A.I., Petrenko, P.B., and Sychev, M.P., *Teoreticheskie osnovy radioelektronnoi razvedki* (Theoretical Foundations of Electronic Reconnaissance), Moscow: Bauman Moscow State Technical University, 2010.
3. Berdyshev, V.P., Garin, E.N., and Fomin, A.N., *Radiolokatsionnye sistemy* (Radar Systems), Krasnoyarsk: Siberian Federal University, 2021.
4. Verba, V.S. and Tatarskii, B.G., *Osnovy teorii radiolokatsionnykh sistem i kompleksov* (Fundamentals of the Theory of Radar Systems and Complexes), Moscow: Tekhnosfera, 2024.
5. Zekavat, S. and Buehrer, R., *Handbook of Position Location: Theory, Practice and Advances*, 2nd ed., Hoboken: Wiley-IEEE Press, 2019.
6. Repin, V.G. and Tartakovskii, G.P., *Statisticheskii sintez pri apriornoi neopredelennosti i adaptatsiya informatsionnykh sistem* (Statistical Synthesis under Prior Uncertainty and Adaptation of Information Systems), Moscow: Sovetskoe Radio, 1977.
7. Zhdanyuk, B.F., *Osnovy statisticheskoi obrabotki traektornykh izmerenii* (Foundations of Statistical Processing of Trajectory Measurements), Moscow: Sovetskoe Radio, 1978.
8. Fomin, A.F., Novoselov, O.N., and Plyushchev, A.V., *Otbrakovka anomal'nykh rezul'tatov izmerenii* (Rejection of Abnormal Measurement Results), Moscow: Energoatomizdat, 1985.

9. Bulychev, Yu.G., Bulychev, V.Yu., Ivakina, S.S., et al., Substantiation of Methods for Optimal Estimation of Target Motion Parameters in Triangulation Location Systems, *J. Comput. Syst. Sci. Int.*, 2015, vol. 54, no. 4, pp. 593–608.
10. Bulychev, V.Y., Bulychev, Y.G., Ivakina, S.S., et al., Classification of Passive Location Invariants and Their Use, *J. Comput. Syst. Sci. Int.*, 2015, vol. 54, no. 6, pp. 905–915.
11. Wang, X., Wang, A., Wang, D., et al., A Modified Sage-Husa Adaptive Kalman Filter for State Estimation of Electric Vehicle Servo Control System, *Energy Reports*, 2022, vol. 8, no. 5, pp. 20–27.
12. Ivanov, A.V., Shishkin, V.Y., Boikov, D.V., et al., Adaptive Algorithms for Information Processing in Navigation Complexes of Mobile Ground Objects, *Journal of Communications Technology and Electronics*, 2021, vol. 66, no. 8, pp. 926–937.
13. Kalianov, A.A., Lukin, O.V., and Tsyganova, I.V., The Algorithm of Adaptive Filtering of Motion Parameters, *Automation of Control Processes*, 2023, no. 1(71), pp. 75–87.
14. Peng, L., Wenhui, W., Junda, Q., et al., Robust Generalized Labeled Multi-Bernoulli Filter and Smoother for Multiple Target Tracking Using Variational Bayesian, *KSII Transactions on Internet and Information Systems*, 2022, vol. 16, no. 3, pp. 908–928.
15. Gao, G., Gao, B., Gao, S., et. al., Hypothesis Test-Constrained Robust Kalman Filter for INS/GNSS Integration with Abnormal Measurement, *IEEE Transact. Vehicular Tech.*, 2023, vol. 72, no. 2, pp. 1662–1673.
16. Granichin, O.N. and Polyak, B.T., *Randomizirovannyye algoritmy otsenivaniya i optimizatsii pri pochti proizvol'nykh pomekhakh* (Randomized Algorithms for Estimation and Optimization under Almost Arbitrary Noise), Moscow: Nauka, 2003.
17. Granichin, O.N., Shalymov, D.S., Avros, R., et al., A Randomized Algorithm for Estimating the Number of Clusters, *Autom. Remote Control*, 2011, vol. 72, no. 4, pp. 754–765.
18. Bulychev, Yu.G. and Chepel', E.N., Optimization of the Cluster-Variant Method of Constructing a Multi-Position Direction Finding System for Conditions of a Priori Uncertainty, *Autom. Remote Control*, 2023, vol. 84, no. 4, pp. 460–473.
19. Tatuzov, A.L., *Neironnye seti v zadachakh radiolokatsii* (Neural Networks in Radiolocation Problems), Moscow: Radiotekhnika, 2009.
20. Stepanov, O.A. and Mostafa, M., Integrated Processing Algorithms for Correction of Navigation System Using Nonlinear Measurements, *Izv. Tula State University*, 2016, no. 6, pp. 89–102.
21. Hajiyev, Ch.M., Abnormal Measurement Rejection for Multidimensional Dynamic Systems, *Avtometriya*, 2003, vol. 39, no. 4, pp. 33–39.
22. Shen, K., Shakhtarin, B.I., Neusypin, K.A., and Nguyen, D.T., Algorithms for Correction of the Navigation Information Using a Satellite Radio Navigation System under Anomalous Measurement Conditions, *Journal of Communications Technology and Electronics*, 2019, vol. 64, no. 1, pp. 29–34.
23. Gu, T., Luo, Z., Guo, T., and Luo, T., A New Reconstruction Method for Measurement Data with Multiple Outliers, *IEEE Transact. Instrument. Measurement*, 2022, vol. 71, art. no. 1005709, pp. 1–9.
24. Ji, C., Song, C., Li, S., et al., An Online Combined Compensation Method of Geomagnetic Measurement Error, *IEEE Sensor. J.*, 2022, vol. 22, no. 14, pp. 14026–14037.
25. Ivanov, A.V., Shishkin, V.Y., Boikov, D.V., et al., Adaptive Algorithms for Information Processing in Navigation Complexes of Mobile Ground Objects, *Journal of Communications Technology and Electronics*, 2021, vol. 66, no. 8, pp. 926–937.
26. Mandel', I.D., *Klasternyi analiz* (Cluster Analysis), Moscow: Finansy i Statistika, 1988.
27. Paklin, N.B. and Oreshkov, V.I., Cluster Silhouettes, *Trudy 20-oi Mezhdunarodnoi nauchno-prakticheskoi konferentsii "Sistemnyi analiz v proektirovanii i upravlenii"* (Proc. Int. Sci.-Pract. Conf. "Systems Analysis in Design and Control"), St. Petersburg, June 29–July 1, 2016, pp. 314–321.

28. Bulychev, Y.G. and Chepel, E.N., Multistructural Method of the Triangulation Estimation of the Motion Parameters of a Radiating Target under A Priori Indefiniteness Assumptions, *J. Comput. Syst. Sci. Int.*, 2019, vol. 58, no. 6, pp. 852–868.
29. Ezhova, N.A. and Sokolinsky, L.B., Survey of Parallel Computation Models, *Bull. South Ural State Univ. Ser. Comput. Math. Software Eng.*, 2019, vol. 8, no. 3, pp. 58–91.
30. Ivanov, A.I. and Shpilevaya, S.G., On Quantum Parallel Computing, *Vestn. Kant Baltic Univ. Ser. Phys., Math., Eng.*, 2021, no. 2, pp. 95–99.

This paper was recommended for publication by N.N. Bakhtadze, a member of the Editorial Board

Program Control of a Multichannel Object Search System with a Finite Number of Processing Lines in Each Channel

V. V. Khutortsev

Research Institute of Radio Communications, Rostov-on-Don, Russia
e-mail: hvv.56@mail.ru

November 28, 2023

February 8, 2025

February 24, 2025

Abstract—The problem of searching for observation objects appearing in accordance with the Poisson flow mathematical model is considered. This problem is solved in the case of a finite number of processing lines in the independent channels of a multichannel search system. Under a high-intensity flow of objects, this may cause queues in object processing. A set of queuing systems with independent incoming flows is selected as a model to determine a search law. The systems of Kolmogorov differential equations describe the dynamics of the probabilistic characteristics of their states under a finite number of processing lines in the search channels. Two formulations of the optimization problem are considered, namely, with probabilistic and time performance criteria. An iterative procedure is proposed to control the distribution of search intensities in the channels of the search system. Examples are given.

Keywords: search for observation objects, multichannel search system, distribution of search intensities, the system of Kolmogorov differential equations

DOI: 10.31857/S0005117925070026

1. INTRODUCTION

The problems of rational effort distribution in search systems (SSs) [1–13] are an important class of problems in the general theory of observation control and management [14–17]. Their effective solution decreases the probability of undetected observation objects (OOs) in the SS coverage area, reduces the time of their search, improves the reliability of OO detection, etc. In particular, such problems include the search for and detection of space debris, since the corresponding flow of particles can be treated as a Poisson flow [18], car traffic monitoring on multilane highways, air traffic management near large airports, airspace use monitoring in megacities, etc.

The rules of solving the above class of problems can be applied to control other systems and queuing networks, including the management of water resources, power grids, data transmission processes in telecommunication systems, etc. [19–26].

The most general structural assumption regarding the mathematical model describing OOs in the search process is their arbitrary (in particular, unlimited) number and appearance in the field of view in accordance with a spatiotemporal random flow [27–30], usually Poisson.

In the latter case, the search problem was investigated in [6–10]. The common feature of the above works is the absence of constraints on the number of processing lines in the SS channels. Under certain conditions, this feature significantly simplified the mathematical models underlying the design of the distribution laws of search intensities. However, for real SSs with a finite number of processing lines in the channels, such distribution law design approaches cannot be used. This is due to the peculiarities of the mathematical models (the Kolmogorov equations) describing the

dynamics of the probabilistic characteristics of the SS channel states in terms of a queuing problem with a finite number of processing lines in each channel [31–34].

In this regard, it is topical to control, via an appropriate law, the distribution of search intensities for multichannel parallel SSs with independent channels and a finite number of processing lines in each channel.

2. STRUCTURAL ANALYSIS OF THE MATHEMATICAL SEARCH MODEL

Let observation objects appear in the SS field of view X in accordance with a spatiotemporal Poisson flow φ [19–22]. Here, $X \subset R^n$ ($1 \leq n \leq 3$), and $(x_1, \dots, x_n)$ denotes the Cartesian frame in X .

Suppose that the field of view X is divided into I zones X_i , $i = \overline{1, I}$, each served by the corresponding i th SS channel.

By assumption, $X = \bigcup_i X_i$, $X_i \cap X_j = \emptyset$, $i = \overline{1, I}$, $j = \overline{1, I}$, $i \neq j$. Then any two flows defined from φ as

$$\varphi_i(t) = \varphi(X_i, t), \quad \varphi_j(t) = \varphi(X_j, t), \quad i = \overline{1, I}, \quad j = \overline{1, I}, \quad i \neq j, \quad (2.1)$$

are Poisson and independent [27, 30].

According to [8, 9, 27, 34, 35], the intensity measures $\xi_i(t)$, $i = \overline{1, I}$, often called the rates of the temporal Poisson flows $\varphi_i(t)$, respectively, are known continuous and bounded deterministic functions. They characterize the temporal Poisson flows $\varphi_i(t)$, $i = \overline{1, I}$, as flows with variable parameters [27–29].

For each zone X_i , the probability of the appearance of one successive OO on a time interval $[t, t + \Delta t]$ is given by [32, 33]

$$\xi_i(t)\Delta t + o(\Delta t), \quad (2.2)$$

whereas the probability of the appearance of more than one OO constitutes $o(\Delta t)$. Here, $o(\Delta t)$ is the residual of an infinitesimal order higher than Δt .

Let us introduce the following constraint: for the i th SS channel, $i = \overline{1, I}$, the number of processing lines is finite and equal to a_i .

We define the search time interval as $\Omega = [0, \bar{t}]$. By assumption, during this interval the i th SS channel provides a search intensity $\lambda_i(t) \geq 0$.

Then, if there are k OOs in the zone X_i , during the time $[t, t + \Delta t]$ at least one of them will be found with the probability [8–10, 32, 33]

$$\begin{aligned} k\lambda_i(t)\Delta t + o(\Delta t), & \quad k \leq a_i, \\ a_i\lambda_i(t)\Delta t + o(\Delta t), & \quad k > a_i. \end{aligned} \quad (2.3)$$

In other words, for $k > a_i$, the probability of detecting an OO on $[t, t + \Delta t]$ stops growing with increasing k .

The search intensities $\lambda_i(t)$ in the zones X_i , $i = \overline{1, I}$, are supposed to be unknown, nonnegative, continuous time-varying functions bounded on $[0, \bar{t}]$. They are to be determined by solving an optimization problem.

The search system problem in the zone X_i can be interpreted as servicing this zone with an incoming OO flow that forms a Poisson-type load with the intensity measure $\xi_i(t)$. As a rule, this class of problems is described using the mathematical apparatus of reproduction and death processes [31–33]. The dynamics of such processes are characterized using infinite-dimensional

systems of Kolmogorov linear ordinary differential equations for the probabilities of process states. These systems form a set of Cauchy problems [32, 33, 36]

$$\begin{aligned}\dot{P}_{i0} &= -\xi_i(t)P_{i0} + \lambda_i(t)P_{i1}, \\ \dot{P}_{ik} &= -(\xi_i(t) + k\lambda_i(t))P_{ik} + \xi_i(t)P_{ik-1} + (k+1)\lambda_i(t)P_{ik+1}, \quad (k = 1, \dots, a_i - 1), \\ \dot{P}_{ik} &= -(\xi_i(t) + a_i\lambda_i(t))P_{ik} + \xi_i(t)P_{ik-1} + a_i\lambda_i(t)P_{ik+1}, \quad k \geq a_i, \\ P_{i0}(0) &= 1, \quad P_{ik}(0) = 0, \quad i = \overline{1, I}, \quad t \in [0, \bar{t}],\end{aligned}\tag{2.4}$$

where P_{i0} is the probability of absence of undetected OOs in X_i or the probability of an event associated with the non-engagement of all processing lines of the i th SS channel for $t \in \Omega$; P_{ik} is the probability of presence of undetected OOs in X_i or the probability of an event associated with the engagement of k processing lines of the i th SS channel if $1 \leq k < a_i$, or a_i processing lines if $k \geq a_i$; the functions $\{\lambda_i(t), \xi_i(t), i = \overline{1, I}\}$ on the interval Ω are supposed to be non-negative, continuous, and bounded.

The Cauchy problem (2.4) reflects the physical meaning of the search carried out by independent channels of the SS with a finite number of processing lines, sequentially in time as the OOs appear in the zones $X_i, i = \overline{1, I}$.

When determining a control law for the distribution of search intensities in the SS, it is difficult to use the mathematical models (2.4) due to their infinite dimension. In addition, the structures of these models eliminate the possibility of their convolution, e.g., in terms of the expected (mean) number of undetected OOs in each SS channel, as was done in [10].

By the service elements (servers) of the i th channel we will understand its processing lines. Then there is no queue in search information processing in the i th channel if:

— None of its servers is busy or none of the processing lines is engaged in processing, which corresponds to $k = 0$.

— k servers ($k = 1, \dots, a_i$) are busy or k processing lines are engaged in processing.

The above states of the i th channel can be represented by the set

$$S_i = \{s_{i0}, s_{i1}, \dots, s_{ik} |_{k=a_i}\}, \quad i = \overline{1, I},\tag{2.5}$$

where each subset s_{ik} ($k = 0, 2, \dots, a_i$) includes $C_{a_i}^k = \frac{a_i!}{k!(a_i-k)!}$ elements.

3. PROBLEM STATEMENT

Let $p_i, i = \overline{1, I}$, denote the probability of an event associated with a single exit of the i th SS channel from the set (2.5) given $0 \leq k \leq a_i$ busy servers at the initial time instant. Note that “single exit” means that the channel leaves S_i exactly once.

Such events are joint and independent, and the probability of their sum is given by the relation [37]

$$p_\Sigma(t) = p_1(t) + p_2(t)(1 - p_1(t)) + \dots + p_I(t) \prod_{i=1}^{I-1} (1 - p_i(t)).\tag{3.1}$$

We define the first performance criterion for the search system as

$$\Upsilon_1 = p_\Sigma(\bar{t}) + \eta \int_0^{\bar{t}} \lambda(t)^T \lambda(t) dt \rightarrow \min_{\lambda \in \Lambda},\tag{3.2}$$

where $\lambda(t) = [\lambda_1(t) \dots \lambda_I(t)]^T$ is the vector of search intensities; $\eta \in R^1$, $\eta > 0$ is a weight coefficient; Λ is a set of admissible control laws (distribution laws of search intensities).

The set of admissible control laws consists of nonnegative and continuous time-varying functions bounded on $[0, \bar{t}]$.

The criterion (3.2) is probabilistic. It assumes minimizing the sum of two components. The first component is the probability with which the state of at least one SS channel will exit the set (2.5) at the terminal time instant $\bar{t}$ of observations. The second component is an analog of the search energy cost of the SS.

Consider the second performance criterion. To do this, we introduce the mean dwell times of the SS channels in the states excluding any queues in search information processing: $m_i = M[\tau_i]$, $i = \overline{1, I}$, where τ_i is a random variable specifying the dwell time of the i th SS channel in the state set (2.5). We define the second performance criterion as

$$\Upsilon_2 = I\bar{t} - \sum_{i=1}^I m_i + \eta \int_0^{\bar{t}} \lambda(t)^T \lambda(t) dt \rightarrow \min_{\lambda \in \Lambda}. \quad (3.3)$$

The criterion (3.3) is a time criterion. It assumes minimizing the sum of two components at the terminal time instant of observations. The first component is the sum of the mean dwell times of the SS channels in the state sets (2.5), taken with a minus sign. The second component, as in (3.2), is an analog of the search energy cost of the SS.

The problem is to design control laws for the distributions of search intensities $\lambda_i(t)$, $i = \overline{1, I}$, $t \in \Omega$, in (2.4) between the channels of the multichannel SS with a finite number of processing lines in each channel that serve the non-intersecting zones X_i , $i = \overline{1, I}$, of the SS field of view X via the probabilistic (3.2) and time (3.3) performance criteria.

4. THE DISTRIBUTIONS OF SEARCH INTENSITIES IN THE MULTICHANNEL SS WITH A FINITE NUMBER OF PROCESSING LINES: CONTROL DESIGN VIA THE PROBABILISTIC CRITERION

Consider the control design procedure for the search for observation objects from the spatiotemporal Poisson flow φ in the multichannel search system with independent channels and a finite number of processing lines in each channel in terms the probabilistic performance criterion (3.2). To do this, we establish the following result.

Proposition 1. *The probability of an event associated with a single exit of the i th SS channel ($i = \overline{1, I}$) from the state set S_i (2.5) provided the belonging of its state to this set at $t = 0$ is given by*

$$p_i(t) = \bar{P}_{i a_i+1}(t), \quad (4.1)$$

where the probability $\bar{P}_{i a_i+1}(t)$ is obtained by solving the system of differential equations

$$\begin{aligned} \dot{\bar{P}}_{i0}(t) &= -\xi_i(t)\bar{P}_{i0}(t) + \lambda_i(t)\bar{P}_{i1}(t), \\ \dot{\bar{P}}_{ik}(t) &= -(\xi_i(t) + k\lambda_i(t))\bar{P}_{ik}(t) + \xi_i(t)\bar{P}_{ik-1}(t) + (k+1)\lambda_i(t)\bar{P}_{ik+1}(t), \quad (k = 1, \dots, a_i - 1), \\ \dot{\bar{P}}_{ia_i}(t) &= -(\xi_i(t) + a_i\lambda_i(t))\bar{P}_{ia_i}(t) + \xi_i(t)\bar{P}_{ia_i-1}(t), \\ \dot{\bar{P}}_{ia_i+1}(t) &= \xi_i(t)\bar{P}_{ia_i}(t), \quad i = \overline{1, I}, \end{aligned} \quad (4.2)$$

with the initial conditions

$$\sum_{k=0}^{a_i} \bar{P}_{ik}(0) = 1, \quad \bar{P}_{i a_i+1}(0) = 0. \quad (4.3)$$

The proof is postponed to the Appendix.

Note that equations (4.2) form a set of fictitious dynamic systems [15] with vector control $\lambda = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_I]^T$. The initial conditions (4.3) determine the belonging of the states of the i th channel to the set S_i at $t = 0$ and correspond to the absence of a queue at its inputs.

Thus, for (3.1) we have

$$p_{\Sigma}(\bar{t}) = \bar{P}_{1 \ a_1+1}(\bar{t}) + \bar{P}_{2 \ a_2+1}(\bar{t})(1 - \bar{P}_{1 \ a_1+1}(\bar{t})) + \dots + \bar{P}_{I \ a_I+1}(\bar{t}) \prod_{i=1}^{I-1} (1 - \bar{P}_{i \ a_i+1}(\bar{t})). \quad (4.4)$$

In view of (3.2) and (4.2), we define the Hamiltonian

$$H = \sum_{i=1}^I \psi_i^T A_i P_i + \eta \lambda^T \lambda, \quad (4.5)$$

where $\psi_i = \psi_i(t) \in R^{a_i+2}$ is the vector of conjugate variables; $P_i(t) = [\bar{P}_{i0}(t) \ \bar{P}_{i1}(t) \ \dots \ \bar{P}_{i \ a_i+1}(t)]^T$; $A_i \in R^{(a_i+2) \times (a_i+2)}$, $A_i = \xi_i B_{i1} + \lambda_i B_{i2}$; the structure of the matrices B_{i1} and B_{i2} follows from (4.2) and is presented in the Appendix.

A peculiarity of the matrices B_{i1} and B_{i2} is the zero sum of each column.

From (4.5) we obtain the systems of equations for $\psi_i(t)$, $i = \overline{1, I}$:

$$\dot{\psi}_i = -\frac{\partial}{\partial P_i} H = -(\xi_i B_{i1}^T + \lambda_i B_{i2}^T) \psi_i, \quad t \in \Omega. \quad (4.6)$$

Due to (4.4), the boundary value conditions for (4.6) can be written as

$$\psi_i(\bar{t}) = \frac{\partial}{\partial P_i(\bar{t})} p_{\Sigma}(\bar{t}) = \left[0 \quad 0 \quad \dots \quad \prod_{j=1, j \neq i}^I (1 - \bar{P}_{j a_j}(\bar{t})) \right]^T, \quad i = \overline{1, I}. \quad (4.7)$$

The minimum condition of (4.5) in λ yields

$$\frac{\partial}{\partial \lambda_i} H = \psi_i^T B_{i2} P_i + 2\eta \lambda_i = 0, \quad i = \overline{1, I}. \quad (4.8)$$

Considering (4.8), we define the optimal control

$$\lambda_i = \frac{-\psi_i^T B_{i2} P_i}{2\eta}, \quad i = \overline{1, I}. \quad (4.9)$$

To solve the two-point boundary value problem (4.2), (4.6), (4.7), we will apply the Krylov–Chernous'ko method of successive approximations [14, 15, 39]. Assume that $\bar{P}_{i0}(0) = 1$ to specify (4.3).

Let a search control law, i.e., a search intensity vector λ^q , be formed at the q th step of the iterative procedure.

The method of successive approximations involves the following operations.

1. Equations (4.2) are solved on the time interval Ω forward, and the vector of variables P_i^q , $i = \overline{1, I}$, corresponding to the distribution of search intensities λ_i^q , $i = \overline{1, I}$, is determined.
2. Based on these solutions, the terminal conditions for the conjugate variables $\psi_i^q(\bar{t})$, $i = \overline{1, I}$, are formed according to (4.7).
3. The system of equations (4.6) is solved on the time interval Ω backward with the terminal conditions (4.7), and the vector of conjugate variables $\psi_i^q(t)$, $t \in \Omega$, $i = \overline{1, I}$, corresponding to the control laws λ_i^q , $i = \overline{1, I}$, and the vector of variables P_i^q , $i = \overline{1, I}$, is determined.

4. Based on the vectors $P_i^q(t)$ and $\psi_i^q(t)$, $i = \overline{1, I}$, the intermediate control vector corresponding to the $(q + 1)$ th step of the iterative procedure is calculated according to (4.9):

$$\tilde{\lambda}_i^{q+1} = \frac{-(\psi_i^q)^T B_{i2} P_i^q}{2\eta}, \quad i = \overline{1, I}, \quad t \in \Omega. \quad (4.10)$$

5. By the principle of partial control update [5, 8, 28], the search intensity control laws for the $(q + 1)$ th iteration are calculated using the distribution of search intensities λ_i^q , $i = \overline{1, I}$, obtained at the previous step:

$$\lambda_i^{q+1}(t) = \varepsilon^q \tilde{\lambda}_i^{q+1}(t) + (1 - \varepsilon^q) \lambda_i^q(t), \quad i = \overline{1, I}, \quad t \in \Omega, \quad (4.11)$$

where $\varepsilon^q \in (0, 1)$.

The parameter ε^q denotes the degree of updating of the search control laws $\lambda_i^q(t)$, $i = \overline{1, I}$, $t \in \Omega$. It is determined by minimizing the objective function of the performance criterion (3.2) at the corresponding step of the iterative procedure.

Next, the control law $\lambda_i^{q+1}(t)$ is used as the initial one for the $(q + 2)$ th step, and operations 1–5 of the iterative procedure are repeated.

Note that for the initial step of the iterative procedure ($q = 0$), the initial distribution of search intensities λ_i^0 , $i = \overline{1, I}$, is selected from the set Λ of admissible control laws.

The optimal control (i.e., the optimal distribution law of search intensities) is given by

$$\lambda_{iOS}(t) = \lim_{q \rightarrow \infty} \lambda_i^q(t), \quad i = \overline{1, I}, \quad t \in \Omega. \quad (4.12)$$

At each step of the iterative procedure, the partial control variation (4.11) obtained by solving the optimization problem is intended to reduce the values of the objective functional. The values $\varepsilon^q \in (0, 1)$ in (4.11) are chosen from the condition of its maximum decrease. On the other hand, according to (3.2), the objective functional corresponding to this criterion is bounded from below. (At least, the functional under consideration cannot be negative.) Thus, the limit (4.12) exists.

In practice, one often performs $q \leq Q$ (finitely many) iterations, where Q is the step number after which the variations of the objective functional of the performance criterion (3.2) become insignificant. By assumption, $\lambda_{iOS}(t) \simeq \lambda_i^Q(t)$, $i = \overline{1, I}$.

5. THE DISTRIBUTIONS OF SEARCH INTENSITIES IN THE MULTICHANNEL SS WITH A FINITE NUMBER OF PROCESSING LINES: CONTROL DESIGN VIA THE TIME CRITERION

Let the upper limit $\bar{t}$ of the observation interval Ω be chosen so that the densities $w_i(t)$, $i = \overline{1, I}$, of the dwell times τ_i of the SS channels in the corresponding sets (2.5) satisfy the conditions

$$\int_0^{\bar{t}} w_i(t) dt = 1, \quad \forall i, i = \overline{1, I}. \quad (5.1)$$

In other words, these densities are finite on Ω .

In view of (5.1), we transform the performance criterion (3.3) as follows.

Assertion 1. *Under condition (5.1), the mean dwell time τ_i of the i th SS channel in the state set S_i (2.5) with the initial conditions (4.3) (a single stay) is given by*

$$m_i = \int_0^{\bar{t}} (1 - \bar{P}_{ia_i+1}(t)) dt. \quad (5.2)$$

The proof of this result is provided in the Appendix.

Considering (5.2), we transform the performance criterion (3.3) to

$$\Upsilon_2 = \int_0^{\bar{t}} \left[\eta \lambda(t)^T \lambda(t) + \sum_{i=1}^I \bar{P}_{ia_i+1}(t) \right] dt \rightarrow \min_{\lambda \in \Lambda}. \quad (5.3)$$

The mathematical models (4.2) and (5.3) are associated with the Hamiltonian

$$H = \sum_{i=1}^I \left(\psi_i^T A_i P_i + \bar{P}_{ia_i+1}(t) \right) + \eta \lambda^T \lambda. \quad (5.4)$$

The equations for the conjugate variables have the form

$$\dot{\psi}_i = -\frac{\partial}{\partial P_i} H = -(\xi_i B_{i1}^T + \lambda_i B_{i2}^T) \psi_i - U_i, \quad i = \overline{1, I}, \quad t \in \Omega, \quad (5.5)$$

where the matrices B_{i1} and B_{i2} , as in the first case, are determined according to (A.2); $U_i \in R^{a_i+2}$, $U_i = [0 \ 0 \ \dots \ 0 \ 1]^T$.

Due to (5.3), the boundary value conditions for (5.5) take the form $\psi_i(\bar{t}) = 0$, $i = \overline{1, I}$; by (5.4), the structure of the resulting control laws coincides with (4.9).

Like for the probabilistic criterion, the two-point boundary value problem (4.2), (5.5) is solved using the Krylov–Chernous'ko successive approximation procedure. It is applied similarly to the previous section.

Owing to the structure of mathematical models in the case under consideration, the computational optimization procedure can be decomposed channel-wise.

If the length of the observation interval is inconsistent with the structure of at least one of the densities $w_i(t)$, $i = \overline{1, I}$ (i.e., equality (5.1) holds only approximately), the solution becomes suboptimal.

6. THE DISTRIBUTIONS OF SEARCH INTENSITIES IN THE MULTICHANNEL SS WITH A FINITE NUMBER OF PROCESSING LINES: CONTROL DESIGN EXAMPLES

Example 1. Consider a two-channel SS ($I = 2$) with three and four processing lines in the first and second channels, respectively. The fictitious dynamic systems corresponding to the SS are described by a mathematical model consisting of two systems of equations (4.2) with $a_1 = 3$ and $a_2 = 4$, respectively. In this case, the performance criterion (3.2) takes the form

$$\Upsilon_1 = \bar{P}_{14}(\bar{t}) + \bar{P}_{25}(\bar{t}) - \bar{P}_{14}(\bar{t}) \bar{P}_{25}(\bar{t}) + \eta \int_0^{\bar{t}} \left(\lambda_1^2(t) + \lambda_2^2(t) \right) dt \rightarrow \min_{\lambda_1, \lambda_2 \in \Lambda}. \quad (6.1)$$

The direct systems of equations (4.2) for $i = 1, 2$ were solved with the initial conditions corresponding to (4.3),

$$\begin{aligned} \bar{P}_{10}(0) &= \bar{P}_{20}(0) = 1, \\ \bar{P}_{11}(0) &= \bar{P}_{12}(0) = \bar{P}_{13}(0) = \bar{P}_{14}(0) = 0, \\ \bar{P}_{21}(0) &= \bar{P}_{22}(0) = \bar{P}_{23}(0) = \bar{P}_{24}(0) = \bar{P}_{25}(0) = 0 \end{aligned}$$

on the search time interval $\Omega = [0, 5]$. From this point onwards, the parameters are presented in dimensionless units.

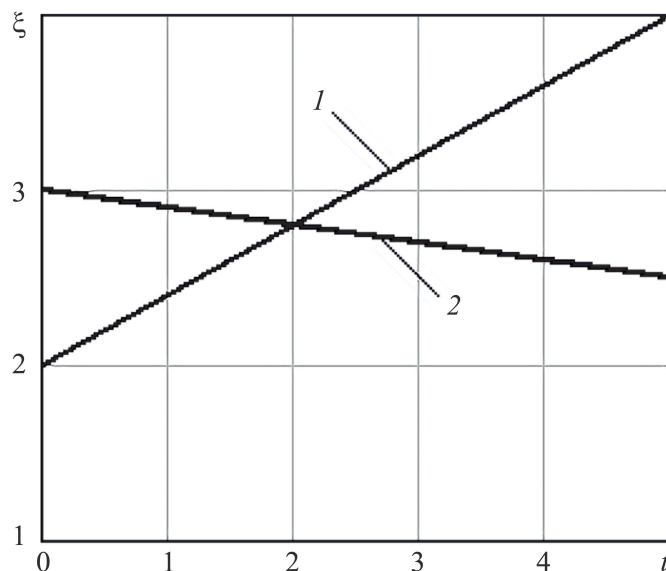


Fig. 1.

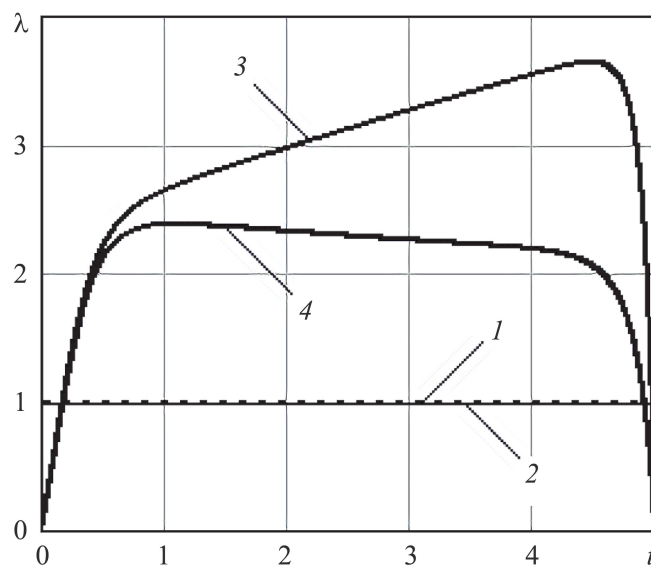


Fig. 2.

The conjugate systems (4.6) are completely determined by the structures of the matrices (A.2) with

$$B_{11}, B_{12} \in R^{5 \times 5} \quad \text{and} \quad B_{21}, B_{22} \in R^{6 \times 6}. \quad (6.2)$$

They were solved in reverse time (backward) with the terminal conditions corresponding to (4.7):

$$\begin{aligned} \psi_1(\bar{t}) &= [0 \ 0 \ 0 \ 0 \ 1 - \bar{P}_{25}(\bar{t})]^T, \\ \psi_2(\bar{t}) &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 - \bar{P}_{14}(\bar{t})]^T. \end{aligned} \quad (6.3)$$

The appearance intensities of OOs in the zones X_1 and X_2 of the SS search area were supposed to be linear functions of time: $\xi_1(t) = 2 + 0.4t$ and $\xi_2(t) = 3 - 0.1t$ (lines 1 and 2 in Fig. 1).

The initial distributions of search intensities λ_1^0 and λ_2^0 were set uniform over the observation interval (lines 1 and 2 in Fig. 2). The equations were solved on the search time interval with a step of $\Delta = 10^{-3}$.

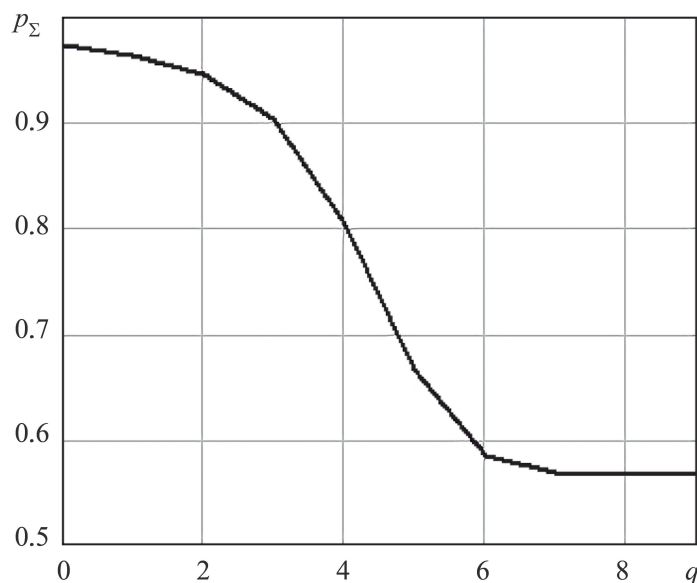


Fig. 3.

Figure 3 shows the probability $p_{\Sigma}(\bar{t})$ of exiting the state sets

$$S_1 = \{s_{10}, s_{11}, \dots, s_{13}\}, \quad S_2 = \{s_{20}, s_{21}, \dots, s_{24}\} \quad (6.4)$$

by at least one of the two SS channels depending on the iteration number q in the Krylov–Chernous’ko successive approximation procedure.

The variations in the probability component of the objective functional of the performance criterion (6.1) become insignificant already for $q \geq Q = 7$. The structure of the optimal control laws for the distribution of search intensities is presented in Fig. 2: curves 3 and 4 for the first and second channels, respectively.

Note that these dependencies are consistent with the variation in the appearance intensities of OOs from the flow (Fig. 1) in the corresponding SS channels.

The relative gain from optimization via the probabilistic criterion makes up $\delta = \frac{p_{\Sigma 0}(\bar{t}) - p_{\Sigma OS}(\bar{t})}{p_{\Sigma 0}(\bar{t})} \simeq 0.42$, where $p_{\Sigma 0}(\bar{t})$ is the probability of exiting the state sets (6.4) by at least one of the two SS channels under λ_1^0, λ_2^0 ; $p_{\Sigma OS}(\bar{t})$ is the same probability under $\lambda_i^Q(t) \simeq \lambda_{iOS}(t)$, $i = 1, 2$.

Example 2. For the two-channel search system of Example 1, consider the control design problem for the distributions of search intensities via the time performance criterion (3.3) in the form (5.3). In this case, the criterion becomes

$$\Upsilon_2 = \int_0^{\bar{t}} [\eta(\lambda_1^2(t) + \lambda_2^2(t)) + \bar{P}_{14}(t) + \bar{P}_{25}(t)] dt \rightarrow \min_{\lambda_1, \lambda_2 \in \Lambda}. \quad (6.5)$$

The direct systems of equations remain unchanged and have the form (4.2). The systems of equations for the conjugate variables correspond to (5.5) with the matrices (6.2), the vectors $U_1 \in R^5$ and $U_2 \in R^6$, and the zero terminal conditions $\psi_i(\bar{t}) = 0$, $i = 1, 2$.

The appearance intensities of OOs in the zones X_1 and X_2 were supposed to be linear functions of time: $\xi_1(t) = 9 + 0.5t$ and $\xi_2(t) = 12 - 0.1t$ (lines 1 and 2, respectively, in Fig. 4).

The initial distributions of search intensities λ_1^0 and λ_2^0 were set uniform over the observation interval (lines 1 and 2 in Fig. 5).

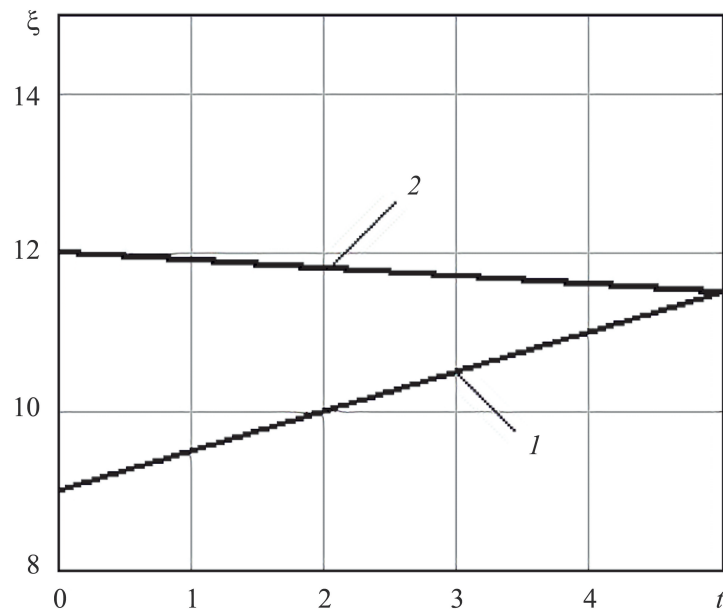


Fig. 4.

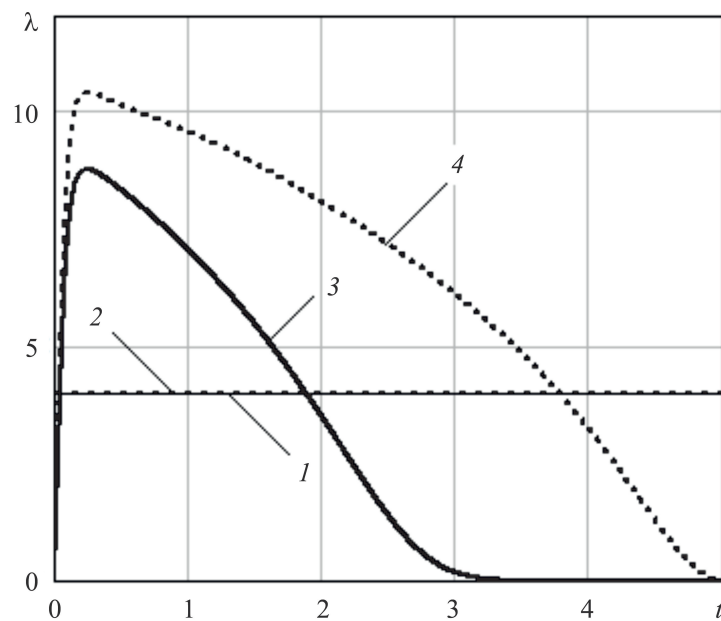


Fig. 5.

Figure 6 shows the mean dwell times m_1 (curve 1) and m_2 (curve 2) of the SS channels are in the state sets (6.4) and their sum (curve 3) depending on the iteration number q in the Krylov–Chernous’ko successive approximation procedure. The time characteristics had insignificant changes for $q \geq Q = 9$.

The smaller growth of m_1 compared to m_2 is due to the smaller number of states in the set S_1 compared to the set S_2 .

The optimal control laws for the distribution of search intensities are demonstrated in Fig. 5: curves 3 and 4 for the first and second SS channels, respectively. The lower search intensity in the first channel is due to the lower appearance intensity of OOs in it (Fig. 4).

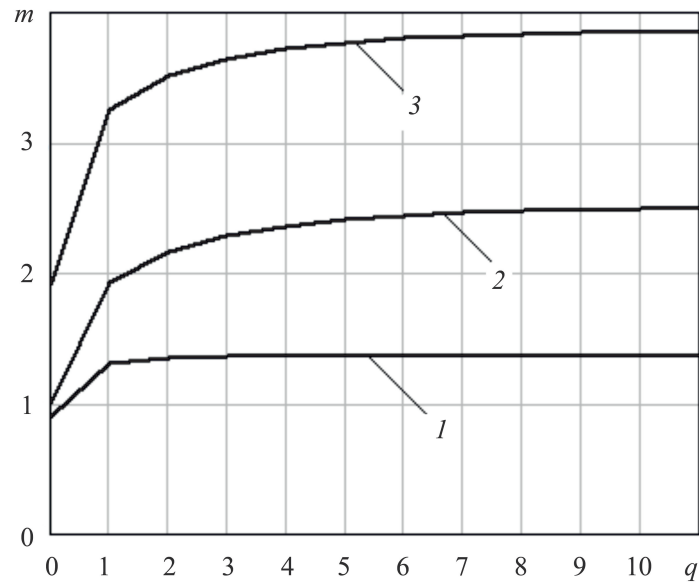


Fig. 6.

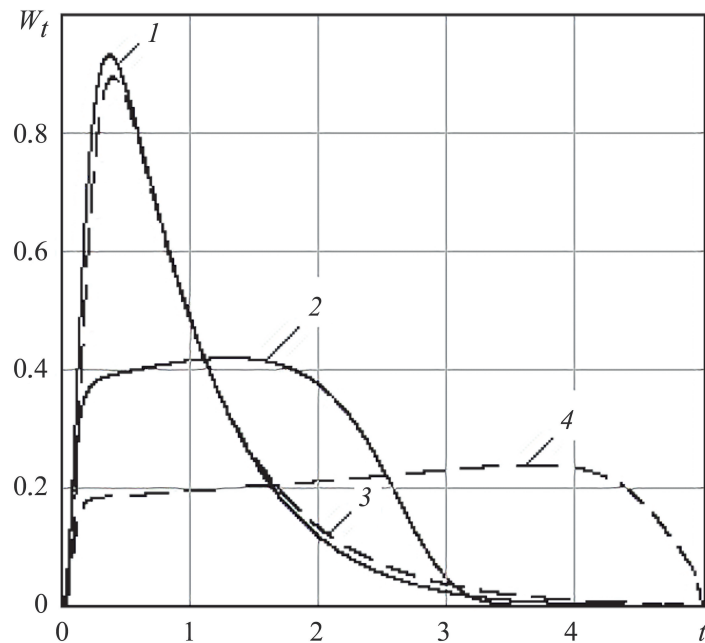


Fig. 7.

Figure 7 presents the structure of the densities of the dwell times of the SS channels in the state sets (6.4): for $t = 0$, curves 1 and 3 for the first and second channels, respectively; for $t = 5$, curves 2 and 4 similarly.

In this example, the optimization procedure increased the total mean dwell time of the SS channels in the state sets (6.4), excluding any queues in search information processing, by more than 2 times.

Considering the structure of the above mathematical models, it is easy to establish the following fact: under the hypotheses of Proposition 1, the search intensities obtained by solving the optimization problems for both performance criteria ((3.2) and (3.3)) always belong to the class of nonnegative, continuous, and bounded functions.

7. CONCLUSIONS

The above control design approach to the search for OOs from a spatiotemporal Poisson flow in a multichannel SS is oriented toward several practically important cases with a finite number of processing lines in each SS channel. In such cases, it is impossible to apply methods associated with the convolution of an infinite system of Kolmogorov equations, e.g., in terms of the mean number of undetected OOs, particularly considered in [10].

The approach is based on the transition from infinite-dimensional systems of differential equations describing the probabilistic characteristics of SS channel states to finite-dimensional auxiliary systems. Their dimensions are determined by the number of processing lines in each channel.

The parameters of the auxiliary systems of equations are the probabilistic characteristics of dwell of the processing lines of SS channels in the state sets excluding any queues for search information processing. With these parameters, it is possible to design control laws for the distribution of search intensities from two viewpoints: minimizing the probability of exiting the above state sets by at least one SS channel or maximizing the mean dwell time of the channels in these sets.

The solution of the first optimization problem with the probabilistic performance criterion (3.2) is more complex due to the structure of the terminal component of this criterion. Characterized by the total dimension of all auxiliary systems of equations (4.2), the dimension of this component is given by $\sum_{i=1}^I a_i + 2I$.

In the second case, the general control design procedure for search intensities can be decomposed channel-wise.

Due to a rather high level of complexity, it is advisable to solve the arising optimization problems using the Krylov–Chernous’ko method of successive approximations together with the principle of partial control update. The examples have shown the possibility of their effective solution using the above approach, particularly the gain from optimization.

Note that the approach can be generalized to the case when the search intensities of OOs in different SS channels are not independent and satisfy the constraint $\sum_{i=1}^I \lambda_i(t) = \lambda_\Sigma = \text{const}$.

APPENDIX

Proof of Proposition 1

We associated with the system of equations (2.4) the state graph [38] presented in Fig. 8.

The graph reflects the sequence of variations in the states of the i th channel with a finite number of processing lines. We select on this graph the states corresponding to the set

$$S_i = \{s_{i0}, s_{i1}, \dots, s_{ik}|_{k=a_i}\} \rightarrow \{\bar{s}_{i0}, \bar{s}_{i1}, b, \dots, \bar{s}_{ik}|_{k=a_i}\} = \bar{S}_i. \quad (\text{A.1})$$

For (A.1), we construct the auxiliary state graph (Fig. 9) as a fragment of the original graph (Fig. 8) with only one absorbing state $\bar{s}_{ia_i+1}$ added.

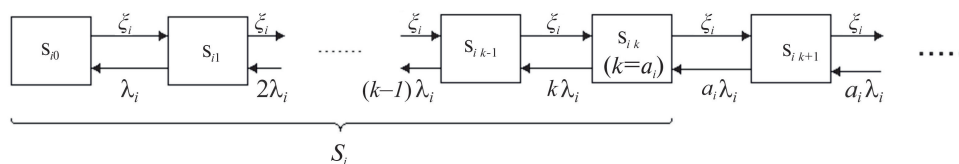


Fig. 8.

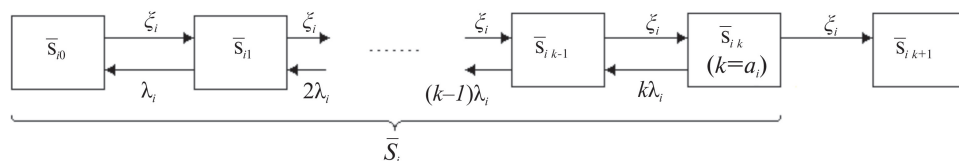


Fig. 9.

It can be shown [38] that the time variation of the probabilistic characteristics $\bar{P}_{i0}(t)$, $\bar{P}_{i1}(t), \dots, \bar{P}_{ia_i}(t), \bar{P}_{ia_i+1}(t)$ of the states $\{\bar{s}_{i0}, \bar{s}_{i1}, \dots, \bar{s}_{ik}|_{k=a_i}, \bar{s}_{ia_i+1}\}$ corresponding to the graph in Fig. 9 is described by the system of differential equations (4.2). For the initial conditions (4.3), the probability of an event associated with a single exit from the set $\bar{S}_i$ (and, accordingly, from the set S_i) is determined by the probability of passing from the state $\bar{s}_{ik}|_{k=a_i}$ to $\bar{s}_{ia_i+1}$. In other words, it corresponds to the component of the solution of system (4.2) determined by the probability $\bar{P}_{ia_i+1}(t)$.

The proof of Proposition 1 is complete.

The structure of the matrices $B_{i1}, B_{i2} \in R^{(a_i+2) \times (a_i+2)}$

$$B_{i1} = \begin{bmatrix} -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad B_{i2} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 2 & 0 & \dots & 0 & 0 \\ 0 & 0 & -2 & 3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_i & 0 \\ 0 & 0 & 0 & 0 & \dots & -a_i & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}. \quad (\text{A.2})$$

Proof of Proposition 2

We write the last equation of the auxiliary system of equations (4.2) corresponding to the i th channel:

$$\dot{\bar{P}}_{ia_i+1}(t) = \xi_i(t)\bar{P}_{ia_i}(t), \quad \bar{P}_{ia_i+1}(0) = 0. \quad (\text{A.3})$$

Recall that the parameter $\bar{P}_{ia_i+1}(t)$ means the probability of an event associated with a single exit of the i th SS channel from the state set S_i (2.5). Therefore, its derivative (A.3) determines the density $w_i(t)$ of the random dwell time τ_i of this channel in the state set S_i . In other words [38],

$$w_i(t) = \xi_i(t)\bar{P}_{ia_i}(t). \quad (\text{A.4})$$

Now we multiply the left- and right-hand sides of (A.3) by t and integrate over the observation interval Ω . As a result, in view of (A.4), we obtain

$$\int_0^{\bar{t}} t \dot{\bar{P}}_{ia_i+1}(t) dt = \int_0^{\bar{t}} t \xi_i(t) \bar{P}_{ia_i}(t) dt = \int_0^{\bar{t}} t w_i(t) dt. \quad (\text{A.5})$$

From (A.5) it follows that

$$\int_0^{\bar{t}} t \dot{\bar{P}}_{ia_i+1}(t) dt = m_i, \quad (\text{A.6})$$

where $m_i = \int_0^{\bar{t}} t w_i(t) dt$ is the mean of the random dwell time τ_i .

Integrating by parts the left-hand side of (A.6) yields

$$t \bar{P}_{ia_i+1}(t) \Big|_0^{\bar{t}} - \int_0^{\bar{t}} \bar{P}_{ia_i+1}(t) dt = m_i. \quad (\text{A.7})$$

Considering condition (5.1), we finally arrive at

$$\int_0^{\bar{t}} (1 - \bar{P}_{ia_i+1}(t)) dt = m_i. \quad (\text{A.8})$$

The proof of Proposition 2 is complete.

REFERENCES

1. Hellman, O., *Vvedenie v teoriyu optimal'nogo poiska* (Introduction to the Theory of Optimal Search), Moscow: Nauka, 1985.
2. Ahlswede, R. and Wegener, I., *Suchprobleme*, Stuttgart: B.G. Teubner, 1979. Translated under the title *Zadachi poiska*, Moscow: Mir, 1985.
3. Abchuk, V.A. and Suzdal', V.G., *Poisk ob"ektov* (Search for Objects), Moscow: Sovetskoe Radio, 1977.
4. Arkin, V.I., A Problem of Optimum Distribution of Search Effort, *Theory Probab. Appl.*, 1964, vol. 9, no. 1, pp. 159–160.
5. Boldyrikhin, N.V. and Khutortsev, V.V., Control of Observations over Random Processes Fluxes, *Autom. Remote Control*, 2006, vol. 67, no. 12, pp. 1900–1912.
6. Khutortsev, V.V. and Baranov, I.V., The Optimization of Observations Control in Problem of Discrete Searching for Poisson Model of the Observed Objects Flow, *Radioengineering*, 2010, no. 3, pp. 20–24.
7. Baranov, I.V. and Khutortsev, V.V., Current Optimization of Search of Objects for the Model of Distributed Poisson Flow of Their Appearance, *J. Comp. Syst. Sci. Int.*, 2011, vol. 50, no. 6, pp. 859–869.
8. Khutortsev, V.V., Optimization of Sequential-Parallel Search for Objects for the Model of the Distributed Poisson Flow of Their Appearance, *J. Comp. Syst. Sci. Int.*, 2019, vol. 58, no. 1, pp. 31–41.
9. Khutortsev, V.V., Controlled Search for Targets Arriving According to a Spatio-Temporal Poisson Point Process by an Information Measurement System with Inhomogeneous Scope, *Autom. Remote Control*, 2021, vol. 82, no. 9, pp. 1568–1580.
10. Khutortsev, V.V., Control of the Search for Observation Objects from a Spatio-Temporal Poisson Flow in a Multi-Channel Search System, *Autom. Remote Control*, 2023, vol. 84, no. 2.
11. Popovich, V.V., Kozhemyaka, V.N., and Grebtsov, G.N., *Teoriya obnaruzheniya i poiska podvizhnykh ob"ektov* (Theory of Detection and Search for Moving Objects), St. Petersburg: Nauka, 2016.
12. Snayd, A. and Stevenson, M., Modelling Stopping Criteria for Search Results Using Poisson Processes, arXiv:1909.06239v1 [cs.IR], September 13, 2019.
13. Filimonov, A.B., Filimonov, N.B., and Tikhonov, V.Yu., Operations Scheduling in Tasks Spatial Search Objects, *Izv. SFedU. Eng. Sci.*, 2017, no. 1–2, pp. 185–197.
14. Chernous'ko, F.L. and Kolmanovskii, V.B., *Optimal'noe upravlenie pri sluchainykh vozmushcheniyakh* (Optimal Control under Random Disturbances), Moscow: Nauka, 1980.
15. Malyshev, V.V., Krasil'shchikov, M.N., and Karlov, V.I., *Optimizatsiya nablyudeniya i upravleniya letatel'nykh apparatov* (Optimization of Surveillance and Control of Aircraft), Moscow: Mashinostroenie, 1989.
16. Grigor'ev, F.N., Kuznetsov, N.A., and Serebrovskii, A.P., *Upravlenie nablyudenyami v avtomaticheskikh sistemakh* (Observation Management in Automatic Systems), Moscow: Nauka, 1986.
17. Miller, B.M., Generalized Optimization in Observation Control Problems, *Autom. Remote Control*, 1991, vol. 52. No. 10. P. 1397–1404.
18. Aslanov, V.S., Pirozhenko, A.V., Voloshenjuk, O.L., Kislov, A.V., and Yashuk, A.V., Definition of a Survival Time of Space Cord System, *Izv. Samara Sci. Cent. Russ. Acad. Sci.*, 2010, vol. 12, no. 4, pp. 138–143.

19. Myasnikov, D.V. and Siemenikhin, K.V. Control of $M|M|1/N$ Queue Parameters under Constraints, *J. Comput. Syst. Sci. Int.*, 2016, vol. 55, no. 1, pp. 59–78.
20. Kurakin, S., Onufrey, A., and Razumov, A., Research of Options for Constructing Information Management Systems Based on Network Models of Queuing Systems, *Informatics and Automation*, 2924, vol. 23, no. 6, pp. 1609–1642.
21. Likhtsinder, B.Ya. and Moiseev, V.I., Batch Poisson and Hyperpoisson Arrival Process Teletraffic Models, *i-Methods*, 2022, vol. 14, no. 3, pp. 1–14.
22. Lichtzinder, B.Y. and Privalov, A.Y., Generalization of Formulas for Queue Length Moments under Nonordinary Poissonian Arrivals for Batch Queues in Telecommunication Systems, *Probl. Inf. Transm.*, 2023, vol. 59, no. 4, pp. 243–248.
23. Butov, A.A. and Galimov, L.A., Optimal Control of Arrival Rate in the Multi-Server Queueing Systems with Router and Occasionally Observed Queue Length on Devices, *Modern Problems of Science and Education*, 2015, no. 2, pp. 758–759.
24. Satrutdinova, A.M., Optimization of Operation of an Electric Grid Enterprise Based on Queuing System, *Vest. Nauki*, 2024, vol. 1, no. 2(71), pp. 97–99.
25. Vasilev, A.P. and Bandurin, I.I., Mathematical Models of Optimum Structure of Operative Service of Electric Networks, *Vest. Ivanovo State Power Eng. Univ.*, no. 2, pp. 1–7.
26. Miller, B.M., Miller, G.B., and Semenikhin, K.V., Methods to Design Optimal Control of Markov Process with Finite State Set in the Presence of Constraints, *Autom. Remote Control*, 2011, vol. 72, no. 2, pp. 571–582.
27. Kingman, J.F.C., *Poisson Processes*, 1st ed., Oxford Studies in Probability, vol. 3, Clarendon Press, 1993.
28. Daley, D.J. and Vere-Jones, D., *An Introduction to the Theory of Point Processes*, New York: Springer, 2013.
29. Last, G., Stochastic Analysis for Poisson Processes, in *Stochast. Anal. for Poisson Point Proc.*, Peccati, G. and Reitzner, M., Eds., Milan: Springer, 2016, pp. 1–36.
30. Khutortsev, V.V., Space-time Poisson Flow Intensity Density with Zero Occurrence Probability on Stochastic Subsets of Its Spatial Definition Domain, *Mathematics and Mathematical Modeling*, 2020, no. 3, pp. 15–28.
31. Khinchin, A.Ya., Mathematical Methods of the Theory of Mass Service, *Tr. Mat. Inst. Steklov.*, 1955, vol. 49, pp. 3–122.
32. Kolmogorov, A.N., The Waiting Problem, in *Teoriya veroyatnostei i matematicheskaya statistika* (Probability Theory and Mathematical Statistics), Moscow: Nauka, 1986, pp. 106–111.
33. Orsingher, T., Cesari, R., and Mosco, V., *An Introduction to Poisson Processes and Their Generalizations*, Instituto Per la Vigilanza Sulle Assicurazioni (IVASS), 2022.
34. Rang, G., Shen, Z., and Zuo, H., Hitting Probabilities of Weighted Poisson Processes with Different Intensities and Their Subordinations, *Acta Math. Sci.*, 2021, vol. 41, no. B(1), pp. 67–84.
35. Feller, W., On Boundaries and Lateral Conditions for the Kolmogorov Differential Equations, *Ann. Math.*, 1957, vol. 65, no. 3, pp. 527–570.
36. Korolyuk, V.S., Portenko, N.I., Skorokhod, A.V., and Turbin, A.F., *Spravochnik po teorii veroyatnostei i matematicheskoi statistike* (Handbook of Probability Theory and Mathematical Statistics), Moscow: Nauka, 1985.
37. Venttsel', E.S. and Ovcharov, L.A., *Teoriya sluchainykh protsessov i ee inzhenernye prilozheniya* (Theory of Random Processes and Its Engineering Applications), Moscow: Nauka, 1991.
38. Krylov, I.A. and Chernous'ko, F.L., On a Method of Successive Approximations for the Solution of Problems of Optimal Control, *USSR Comp. Math. Math. Phys.*, 1963, vol. 2, no. 6, pp. 1371–1382.

This paper was recommended for publication by A.I. Kibzun, a member of the Editorial Board

Developing a Multi-Output Soft Sensor Based on an Error Predictor for Quality Control of the Target Product of a Complex Distillation Column

D. V. Shtakin^{*,a}, A. A. Plotnikov^{*,b}, O. Yu. Snegirev^{*,c}, and A. Yu. Torgashov^{*,d}

Institute of Automation and Control Processes, Far Eastern Branch,

Russian Academy of Sciences, Vladivostok, Russia

e-mail: ^ashtakin@iacp.dvo.ru, ^bplotnikov_aa@iacp.dvo.ru,

^csnegirevoleg@iacp.dvo.ru, ^dtorgashov@iacp.dvo.ru

Received January 31, 2025

Revised April 30, 2025

Accepted May 5, 2025

Abstract—This paper addresses the problem of improving the accuracy of a multi-output soft sensor (SS). It is demonstrated that implementing a predictor for the vector time series allows for consideration of the dynamic interdependence of process components, thereby enhancing the SS accuracy. The development of a multi-output error predictor is performed using vector autoregressive models and a set of autoregressive distributed lag models, with their optimal structures and parameters determined by numerical methods. The proposed approach to developing a multi-output SS is compared with traditional methods based on the sequential development of single-output SS within the quality control system of the target product (light diesel fraction) of a complex industrial distillation column. The effectiveness of the proposed approach is also demonstrated for the class of adaptive SS.

Keywords: multi-output soft sensor, multi-output error predictor, vector autoregressive model, autoregressive distributed lag model, adaptation, complex distillation column

DOI: 10.31857/S0005117925070038

1. INTRODUCTION

Soft sensors (SS) are key elements of modern process control systems, responsible for providing feedback on the quality of the final product [1]. Fractional distillation (or fractionation) is one of the most common and energy-intensive continuous technological processes in oil refining and the petrochemical industry. In industrial fractionating columns, crude oil feedstock is separated into various fractions of petroleum products. Typically, the separation occurs in complex columns featuring several intermediate circulating refluxes and the withdrawal of multiple products. The target, or most valuable, product is identified, and stabilizing its quality provides significant economic benefits for the production facility. Important quality indicators for the target product, such as the light diesel fraction of a hydrocracking process unit, are the key points of the fractional composition (initial boiling points, 10, 50, 90, 95%). These are determined in a plant laboratory at a frequency of once or twice per day, which is insufficient for real-time optimal control tasks. Therefore, the implementation of a SS primarily serves to estimate the quality indicators of the products at each control step. A SS typically includes statistical models that correlate hard-to-measure quality indicators (output variables) with easily accessible measurements (input variables) of process parameters (temperature, pressure, flow rates, etc.). The efficiency of process control largely depends on the accuracy of the models within the SS. In this regard, improving the accuracy of the SS is crucial and directly related to enhancing the efficiency of production process

control [2, 3]. It is worth noting that the task of identifying dependencies between the quality indicators of fractionating column products and a set of input variables has been extensively studied for the case of a single-output variable [4–8]. Methods for developing SS based on neural networks have also become widespread. Study [9] is devoted to the development of a dynamic soft sensor based on a convolutional neural network that accounts for changes in the characteristics of the analyzed process over time. The use of deep learning methods for developing SS is also known [10]. The work [11] focuses on a neural network training method combining backpropagation with partial least squares. The key advantage of neural network methods is their ability to approximate nonlinear dependencies with high accuracy on the training sample; however, significant errors can be observed on the test sample.

It should be noted that the task of developing a multi-output SS (MOSS) is highly relevant. In comparison with traditional single-output models, multi-output regression analysis better explains the dependency between the evaluated variables and the input data, due to the correlation among the output variables when considering complex relationships between the input and output data [12]. The problem of developing a MOSS can be addressed by developing new methods and models, as in [13], which describes a method for developing a multi-output SS based on multi-output tree chaining. The essence of this method lies in constructing an ensemble of trees that evaluate several output variables simultaneously. Existing methods can also be improved for developing a MOSS. For instance, [14] presents a modification of the Least-Squares Support Vector Machine method (LS-SVM) for developing models with multiple outputs and accounting for the nonlinear relationship between different output data, along with a proposed training method for such a model.

A widely adopted class of methods for improving SS accuracy for non-stationary plants involves the development of adaptive SS. The paper [15] is dedicated to the development of ensemble adaptive SS based on several Gaussian process models. In [16], the combination of several adaptation mechanisms to achieve the best SS accuracy is discussed. Methods for developing adaptive SS based on the Just-In-Time (JIT) approach are also common [17]. The most significant advantage of methods for developing adaptive soft sensors is the ability to account for process evolution over time, not only at the level of updating the model parameters but also at the level of updating the model structure, as shown in works [18, 19] devoted to SS based on the JIT approach combined with online selection of SS input variables. It should be noted that the effectiveness of the JIT approach decreases with less data in the training sample, which is a significant limitation in real-world conditions and prevents its use for developing MOSS in the presence of significant data gaps or small training samples.

There is also a class of methods for improving the accuracy of SS through bias updates (BU) of the regression model. In [20], a strategy for updating the bias term of the regression model is described, based on a Bayesian approach that considers the mean value and standard deviation of the SS error. A weighted strategy for BU of the regression model is known from [21]. This strategy is based on considering the previous values of the regression model's bias term.

The common drawback of the above methods and approaches is their failure to account for the mutual influence between the evaluated quality indicators, which in some cases can be of significant importance. There are a number of works treating the task of BU as the task of predicting the SS error, which allows accounting for the dynamic influence of the error on SS accuracy. For example, in [22], the bias term update strategy for the MOSS regression model is considered as the task of predicting the soft sensor error using a moving-average autoregressive model. The work [23] is devoted to the development of a predictive filter for adaptive SS. It should be noted that the works [19–22] do not account for the mutual influence of correlated SS errors. In [24], the problem of predicting the vector of cross-correlated SS errors is solved using analytical methods. However,

the choice of the predictor model order from auto- and cross-spectral forms does not allow for structural adaptation of the predictor in online mode at each discrete time step, and also increases the complexity of developing SS for the case of n -outputs.

This paper proposes a method for developing a MOSS with a predictor, distinguished by taking into account the interdependence of errors using vector autoregressive (VAR) models [25] or autoregressive distributed lag (ARDL) models [26]. These models are developed using numerical methods, which allows for both structural and parametric adaptation of the SS error predictor at each discrete moment in time in a closed loop of the quality control system for the target product of a continuous technological process.

2. PLANT DESCRIPTION AND PROBLEM STATEMENT

The object of the study is a complex distillation column of a hydrocracking process unit for separating the feed into the following products: gasoline fraction (GF), kerosene fraction (KF), light diesel fraction (LDF), heavy diesel fraction (HDF), and hydrocracking residue (HR). LDF is subsequently used as a component of arctic diesel fuel (ADF). Figure 1 shows the flow chart of the distillation process and the control system with a controller based on the predictive model (PM) for the LDF fractional composition. The following conventions are used: K1 – complex distillation

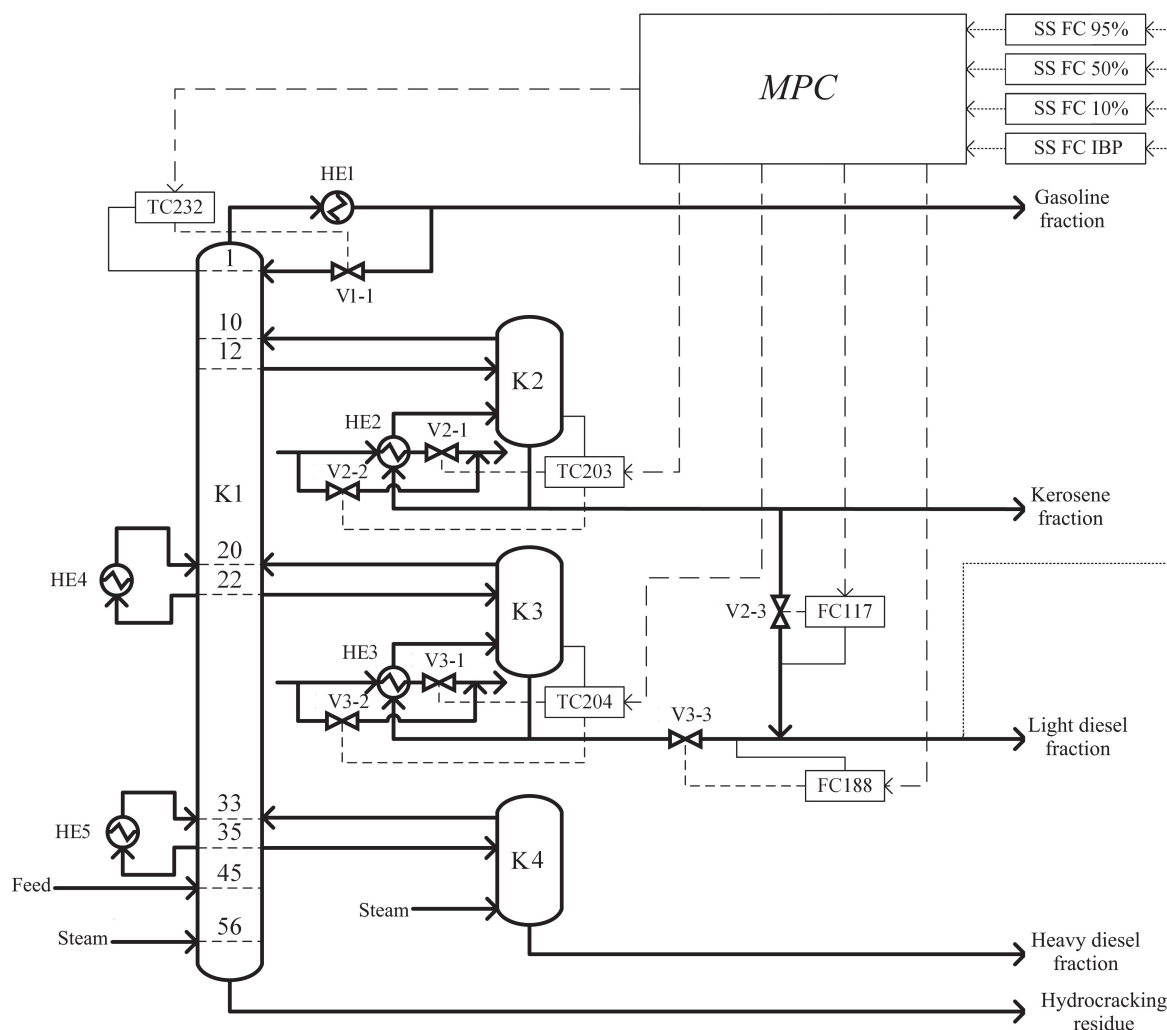


Fig. 1. Industrial plant scheme and control system.

column; K2 – kerosene fraction stripping column; K3 – diesel fraction stripping column; K4 – heavy diesel fraction stripping column; HE1 – gasoline fraction vapor condenser; HE2 – reboiler of column K2, HE3 – reboiler of column K3, HE4 – cooler of upper circulation reflux; HE5 – cooler of lower circulation reflux; V1-1 – valve on bypass of HE1; V2-1 – valve on outlet of HE2; V2-2 – valve on bypass of HE2; V2-3 – valve on flow line of KF for mixing with LDF; V3-1 – valve on outlet of HE3; V3-2 – valve on bypass of HE3; V3-3 – valve on flow line of LDF for mixing with KF. Table 1 shows a fragment of matrix of gain factors of process object. Table 2 shows the input variables of the fractional composition (FC) of the LDF used in industrial conditions: FC initial boiling point (IBP), 10, 50, 95%.

Table 1. Fragment of the gain factors matrix of an industrial plant

SS	TC232 (K1 top temp.)	TC203 (K2 temp.)	TC204 (K3 temp.)	FC117 (KF rate for mixing with LDF)	FC188 (LDF rate for mixing with KF)
FC IBP	2,2	2,4	2,7	0,5	
FC 10%	1,1	1,2	2,3	0,35	
FC 50%				0,2	1,2
FC 90%					1,7

Table 2. MOSS input variables

No.	Designation	Description	Unit of meas.
1	P131	K1 top pressure	MPa
2	P124	K1 bot pressure	MPa
3	FC105	K1 reflux rate	m ³ /h
4	FC106	K1 top pumparound stream	m ³ /h
5	FC107	K1 bottom pumparound stream	m ³ /h
6	FC116	KF product rate	m ³ /h
7	FC117	KF rate for mixing with LDF	m ³ /h
8	FC188	LDF rate for mixing with KF	m ³ /h
9	TC232	K1 top temperature	°C
10	T233	Return upper product temperature from K2 to K1	°C
11	T234	Return upper product temperature from K3 to K1	°C
12	T235	Return upper product temperature from K4 to K1	°C
13	T238	Sidestream temperature from K1 to K2	°C
14	T239	Sidestream temperature from K1 to K3	°C
15	T240	Sidestream temperature from K1 to K4	°C
16	T242	Temperature above the feedstock input zone in K1	°C
17	T247	Temperature at the outlet of HE2 into K2	°C
18	T248	Temperature at the outlet from K2 to HE2	°C
19	T253	Temperature at the outlet from HE3 to K3	°C
20	T254	Temperature at the outlet from K3 to HE3	°C
21	TC275	Temperature of bottom pumparound to K1	°C
22	TC279	Temperature of top pumparound to K1	°C
23	FY098	K1 feed rate	m ³ /h

The problem of estimating the parameters of the MOSS and constructing a model of correlated error predictor of the SS is considered. In general, the MOSS model with a correlated error predictor

(MOSS with a CEP) has the following form:

$$\hat{\mathbf{Y}}_t^* = K \mathbf{X}_t + \tilde{\mathbf{b}} + \Phi \mathbf{E}_t^{past}, \quad (1)$$

where $\hat{\mathbf{Y}}_t^* = (\hat{y}_{1,t}^* \dots \hat{y}_{j,t}^* \dots \hat{y}_{N,t}^*)^T$ is a vector of estimated values of quality indicators;

$$K = \begin{pmatrix} k_{1,1} & \dots & k_{1,j} & \dots & k_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ k_{h,1} & \dots & k_{h,j} & \dots & k_{h,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ k_{N,1} & \dots & k_{N,j} & \dots & k_{N,m} \end{pmatrix} \text{ is a MOSS coefficient matrix; } \mathbf{X}_t = (x_{1,t} \dots x_{j,t} \dots x_{m,t})^T$$

is MOSS input variables vector; $\tilde{\mathbf{b}} = (b_1 \dots b_h \dots b_N)^T$ is a MOSS free term vector; m is a number of MOSS input variables; N is a number of MOSS output variables;

$$\Phi = \begin{pmatrix} \phi_{1,1|1} & \dots & \phi_{1,1|\alpha} & \dots & \phi_{1,N|1} & \dots & \phi_{1,N|\alpha} \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ \phi_{N,1|1} & \dots & \phi_{N,1|\alpha} & \dots & \phi_{N,N|1} & \dots & \phi_{N,N|\alpha} \end{pmatrix} \text{ is a coefficient matrix of MOSS correlated}$$

error predictor; $\phi_{i,j|\alpha}$ is α th MOSS correlated error predictor coefficient for considering of the influence between the j th and i th components of the multivariable error series; $\mathbf{E}_t^{past} = (e_{1,t-1} \dots e_{1,t-\alpha} \dots e_{N,t-1} \dots e_{N,t-\alpha})^T$ is an error predictor input vector; α is an error predictor order; $e_{i,t} = y_{i,t} - (b_i + [K]_i \mathbf{X}_t)$ is a MOSS estimation error of the i th quality indicator at time t ; $[\cdot]_i$ is an i th matrix row.

MOSS with a preliminary whitening correlated error predictor (i.e. with passage through a filter that equalizes the spectral density) of an error series using the AR model (MOSS with a WCEP) is described by the expression:

$$\hat{\mathbf{Y}}_t^{**} = K \mathbf{X}_t + \tilde{\mathbf{b}} + W \mathbf{E}_t^{past} + \tilde{\Phi} \mathbf{V}_t^{past}, \quad (2)$$

where $\mathbf{E}_t^{past} = (e_{1,t-1} \dots e_{1,t-q} \dots e_{N,t-1} \dots e_{N,t-q})^T$ is a MOSS error vector;

$$W = \begin{pmatrix} w_{1,1|1} & \dots & w_{1,1|q} & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & w_{2,2|1} & \dots & w_{2,2|q} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & w_{N,N|1} & \dots & w_{N,N|q} \end{pmatrix} \text{ is a whitening pre-}$$

dictor coefficient matrix with order q ; $w_{i,j|q}$ is a q th whitening predictor coefficient for considering the influence of the j th element on the i th multidimensional error series;

$$\tilde{\Phi} = \begin{pmatrix} \tilde{\phi}_{1,1|1} & \dots & \tilde{\phi}_{1,1|\tilde{\alpha}} & \dots & \tilde{\phi}_{1,N|1} & \dots & \tilde{\phi}_{1,N|\tilde{\alpha}} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \tilde{\phi}_{N,1|1} & \dots & \tilde{\phi}_{N,1|\tilde{\alpha}} & \dots & \tilde{\phi}_{N,N|1} & \dots & \tilde{\phi}_{N,N|\tilde{\alpha}} \end{pmatrix} \text{ is a whitening correlated error predictor}$$

coefficient matrix with order $\tilde{\alpha}$; $\mathbf{V}_t^{past} = (v_{1,t-1} \dots v_{1,t-\tilde{\alpha}} \dots v_{N,t-1} \dots v_{N,t-\tilde{\alpha}})^T$ is an error predictor input vector; $v_{i,t} = e_{i,t} - [W]_i \mathbf{E}_t^{past}$ is a whitening error for i th quality indicator MOSS error.

The required parameters matrix θ_1 for (1) and θ_2 for (2) have the following form:

$$\theta_1 = (K \ \tilde{\mathbf{b}} \ \Phi), \quad (3)$$

$$\theta_2 = (K \ \tilde{\mathbf{b}} \ W \ \tilde{\Phi}). \quad (4)$$

The determination of estimates of unknown parameters θ_z ($z = 1, 2$) is carried out by solving an optimization problem for a given quality functional J :

$$\hat{\theta}_z = \underset{\theta_z \in \mathbb{R}}{\operatorname{argmin}} J(\theta_z). \quad (5)$$

The MOSS mean square error is considered as the criterion J :

$$J = \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{Y}_i - \hat{\mathbf{Y}}_i(\theta_z) \right\|^2, \quad (6)$$

where $\mathbf{Y}_i$ and $\hat{\mathbf{Y}}_i$ are the vectors of measured and estimated values of the target product quality indicators, respectively; n is an observations number in a sample; $\|\cdot\|$ is an Euclidean norm.

3. DEVELOPING A MULTI-OUTPUT SOFT SENSOR

The diagram of the proposed multi-output soft sensor with a predictor as part of the control system is shown in Fig. 2, where MPC is a model predictive controller; $G(U)$ is an industrial plant; U is the control actions vector; $\hat{\mathbf{Y}}_t$ is a SS estimate (no consideration of error predictor); $\mathbf{e}_t$ is a vector of SS errors; $\mathbf{v}_t$ is a vector of whitening errors; $\hat{\mathbf{e}}_t$ is an error prediction; $\hat{\mathbf{v}}_t$ is a whitening error prediction; B is the back shift operator. The main advantage of the proposed multi-output soft sensor lies in accounting for the dynamic influence of correlated errors in evaluating the quality indicators of the target product of a complex fractionating column. It should be noted that the proposed predictor is an add-on to any model within the multi-output soft sensor and can be used in combination with other models, such as neural network models that form the basic component of the yield estimation $\hat{\mathbf{Y}}_t$.

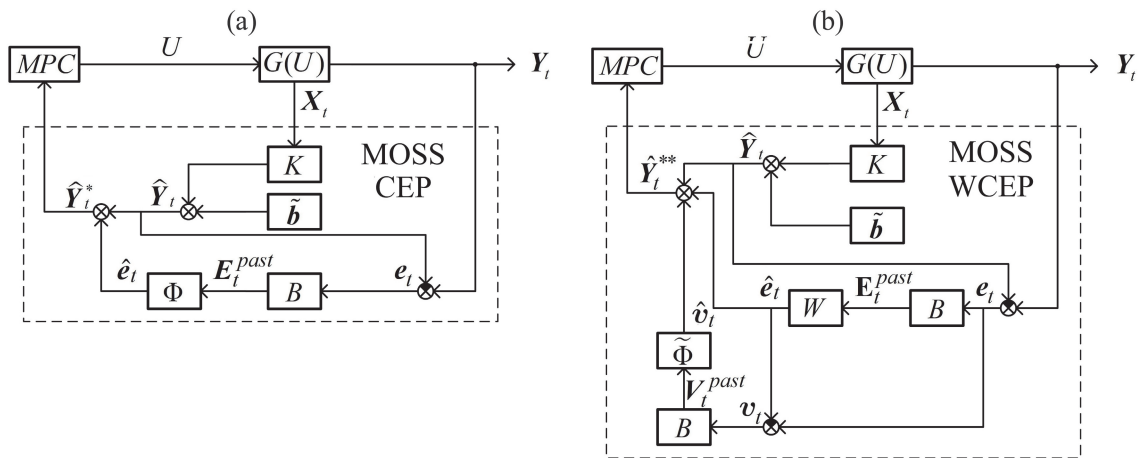


Fig. 2. Structural scheme of the control system with MOSS with a predictor of correlated errors: (a) without preliminary whitening; (b) with preliminary whitening.

In expressions (1) and (2) the first two terms $(K\mathbf{X}_t + \tilde{\mathbf{b}})$ represent the multiple linear regression (MLR) model, which is widely used in industrial settings [27].

The search for a solution to the optimization problem (5) consists of several stages. At the first stage, the first two blocks of the matrix θ_z ($\hat{K}$ and $\hat{\tilde{\mathbf{b}}}$), which are responsible for the model within the MOSS, are determined:

$$\{\hat{K}, \hat{\tilde{\mathbf{b}}}\} = \underset{K, \tilde{\mathbf{b}} \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{Y}_i - (K\mathbf{X}_i + \tilde{\mathbf{b}}) \right\|^2. \quad (7)$$

At the second stage, the MOSS error predictor is developed, i.e. the third block of matrices θ_z is determined. At this stage, for the multi-output soft sensor with a correlated error predictor (CEP), $\hat{\Phi}$ is determined taking into account the cross-correlations of the MOSS errors:

$$\hat{\Phi} = \operatorname{argmin}_{\Phi \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{e}_i - \Phi \mathbf{E}_i^{\text{past}} \right\|^2. \quad (8)$$

For the MOSS with a whitening correlated error predictor (WCEP), $\hat{W}$ is determined, i.e. the parameters of the “whitening” predictor are determined without taking into account the cross-correlations of the MOSS errors:

$$\hat{W} = \operatorname{argmin}_{W \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{e}_i - W \mathbf{E}_i^{\text{past}} \right\|^2. \quad (9)$$

Since the interdependence of the error series is not accounted for in the MOSS with a WCEP in the previous stage, a third stage becomes necessary, during which the optimal parameters of the “whitening” error predictor $\hat{\tilde{\Phi}}$ are determined, taking their cross-correlation into account:

$$\hat{\tilde{\Phi}} = \operatorname{argmin}_{\tilde{\Phi} \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{v}_i - \tilde{\Phi} \mathbf{V}_i^{\text{past}} \right\|^2. \quad (10)$$

Well-known autoregressive (AR) models can be used as a model of the “whitening” predictor W . Typically, single-output AR models make it possible to significantly increase the SS accuracy. However, this approach has its drawbacks, namely, it does not account for the correlation of SS errors for interconnected quality indicators.

The MOSS with a CEP described in (1) is further considered for the case $N = 2$, and its parameters θ_1 are found using (7) and (8). In this paper, the CEP for the MOSS is implemented based on the following VAR model:

$$\begin{pmatrix} \hat{e}_{1,t} & \hat{e}_{2,t} \end{pmatrix}^T = \Omega \mathbf{E}_t^{(q)}, \quad (11)$$

where $\Omega = \begin{pmatrix} \omega_{1,1|1} & \dots & \omega_{1,1|j} & \dots & \omega_{1,1|q} & \omega_{1,2|1} & \dots & \omega_{1,2|j} & \dots & \omega_{1,2|q} \\ \omega_{2,1|1} & \dots & \omega_{2,1|j} & \dots & \omega_{2,1|q} & \omega_{2,2|1} & \dots & \omega_{2,2|j} & \dots & \omega_{2,2|q} \end{pmatrix}$ is the coefficients matrix of the VAR model; $\mathbf{E}_t^{(q)} = \begin{pmatrix} e_{1,t-1} & \dots & e_{1,t-j} & \dots & e_{1,t-q} & e_{2,t-1} & \dots & e_{2,t-j} & \dots & e_{2,t-q} \end{pmatrix}^T$ is the input vector of the VAR model; q is VAR model order.

This paper also considers another method for implementing the CEP using an ARDL model [26] of the following form:

$$\hat{e}_{1|t} = \mathbf{a}_1 E_{1|t}^{\text{endo}(1)} + \mathbf{l}_1 E_{2|t}^{\text{exo}(1)}, \quad \hat{e}_{2|t} = \mathbf{a}_2 E_{2|t}^{\text{endo}(2)} + \mathbf{l}_2 E_{1|t}^{\text{exo}(2)}, \quad (12)$$

where $\mathbf{a}_1 = (a_{1,1} \dots a_{1,j} \dots a_{1,A_1})$ and $\mathbf{a}_2 = (a_{2,1} \dots a_{2,j} \dots a_{2,A_2})$ are the regression coefficients of the endogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; $\mathbf{l}_1 = (l_{1,1} \dots l_{1,j} \dots l_{1,L_1})$ and $\mathbf{l}_2 = (l_{2,1} \dots l_{2,j} \dots l_{2,L_2})$ are the regression coefficients of the exogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; $E_{1|t}^{\text{endo}(1)} = (e_{1,t-1} \dots e_{1,t-j} \dots e_{1,t-A_1})^T$ and $E_{2|t}^{\text{endo}(2)} = (e_{2,t-1} \dots e_{2,t-j} \dots e_{2,t-A_2})^T$ are the input vectors of the endogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; $E_{2|t}^{\text{exo}(1)} = (e_{2,t-d_1} \dots e_{2,t-d_1-j} \dots e_{2,t-d_1-L_1+1})^T$ and

$E_{1|t}^{exo(2)} = (e_{1,t-d_2} \dots e_{1,t-d_2-j} \dots e_{1,t-d_2-L_2+1})^T$ are the input vectors of the exogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; A_1 and A_2 are the regression orders of the endogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; d_1 and d_2 are the lags of the exogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$; L_1 and L_2 are the regression orders of the exogenous component of the model error series $\hat{e}_1$ and $\hat{e}_2$.

The ARDL model is a more flexible tool for accounting for the correlation of SS errors in comparison with the VAR model, since it makes it possible to set different regression orders for endogenous and exogenous components, as well as the lag value for the exogenous component. The proposed approach to developing a SS includes the possibility of preliminary whitening (2) of the SS error series (correction of the SS error using the AR model) before applying the VAR or ARDL models (MOSS with a WCEP). The parameters of the MOSS with a WCEP are found using (7), (9), (10). In this case, the VAR and ARDL models are used to forecast not the SS errors but the whitening errors:

$$v_{1,t} = e_{1,t} - \hat{e}_{1,t}, \quad v_{2,t} = e_{2,t} - \hat{e}_{2,t}. \quad (13)$$

Therefore, the VAR (11) and ARDL (12) models take the form (14) and (15), respectively:

$$\begin{pmatrix} \hat{v}_{1,t} & \hat{v}_{2,t} \end{pmatrix}^T = \Omega \mathbf{V}_t^{(q)}, \quad (14)$$

where $\mathbf{V}_t^{(q)} = (v_{1,t-1} \dots v_{1,t-j} \dots v_{1,t-q} \quad v_{2,t-1} \dots v_{2,t-j} \dots v_{2,t-q})^T$ is a VAR model input vector.

$$\hat{v}_{1|t} = \mathbf{a}_1 V_{1|t}^{endo(1)} + \mathbf{l}_1 V_{2|t}^{exo(1)}, \quad \hat{v}_{2|t} = \mathbf{a}_2 V_{2|t}^{endo(2)} + \mathbf{l}_2 V_{1|t}^{exo(2)}, \quad (15)$$

where $V_{1|t}^{endo(1)} = (v_{1,t-1} \dots v_{1,t-j} \dots v_{1,t-A_1})^T$ and

$V_{2|t}^{endo(2)} = (v_{2,t-1} \dots v_{2,t-j} \dots v_{2,t-A_2})^T$ are the input vectors of the endogenous component of the model error series $\hat{v}_1$ and $\hat{v}_2$;

$V_{2|t}^{exo(1)} = (v_{2,t-d_1} \dots v_{2,t-d_1-j} \dots v_{2,t-d_1-L_1+1})^T$ and

$V_{1|t}^{exo(2)} = (v_{1,t-d_2} \dots v_{1,t-d_2-j} \dots v_{1,t-d_2-L_2+1})^T$ are the input vectors of the exogenous component of the model error series $\hat{v}_1$ and $\hat{v}_2$.

As a rule, the order of an AR model for a time series is determined by visual analysis of its correlogram or autospectrum. Despite the wide prevalence of this approach, in the present work, a numerical method is used to determine the AR model order, based on the analysis of the partial autocorrelation estimate $\hat{\psi}_k$, determined by the Yule-Walker equation:

$$\hat{\Psi}^{(k)} = P_k^{-1} \rho^{(k)}, \quad (16)$$

where k is the lag; $\hat{\Psi}^{(k)} = (\hat{\psi}_1 \dots \hat{\psi}_j \dots \hat{\psi}_k)^T$ is an AR model coefficients vector with order k ; $\rho^{(k)} = (r_1 \dots r_j \dots r_k)^T$ is an autocorrelation vector of estimates before lag k ;

$P_k = \begin{pmatrix} 1 & r_1 & r_2 & \dots & r_{k-1} \\ r_1 & 1 & r_1 & \dots & r_{k-2} \\ r_2 & r_1 & 1 & \dots & r_{k-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{k-1} & r_{k-2} & r_{k-3} & \dots & 1 \end{pmatrix}$ is the Toeplitz matrix based on $\rho^{(k)}$. The equation (16) links

the coefficients of the AR model and the values of the autocorrelation estimate [28]. The use of

numerical methods allows for the automation of the AR model order determination process and the implementation of structural predictor adaptation. The method for determining the AR model order based on the analysis of the estimate $\hat{\psi}_k$ is as follows. The value of the partial autocorrelation estimate at lag k is set to zero if $|\hat{\psi}_k| < 2\hat{\sigma}_{\hat{\psi}_k}$, where $\hat{\sigma}_{\hat{\psi}_k}$ is the standard error of the partial autocorrelation estimates, defined as $\hat{\sigma}_{\hat{\psi}_k} = \frac{1}{n}$. Then, the maximum lag value k for which $\hat{\psi}_k \neq 0$ is used as the order p .

Partial autocorrelation is also used to determine the order q of the VAR model. Using the approach described above, the orders of the AR models for each error series included in the vector model are determined, after which the smallest of the previously obtained AR model orders is chosen as the vector model order q . This approach helps to avoid including redundant variables in the model.

To determine the values of A and L for the ARDL model, the method from [29] can be used, which is based on identifying the instability of the figures of the model parameter estimates as the model structure becomes more complex. A sequential increase in the model order is performed as long as the following condition is met:

$$\sum_{i=0}^{q-1} \text{sgn } b_i^{(q-1)} \text{sgn } b_i^{(q)} = q - 1, \quad (17)$$

where q is order of model: $\hat{x}_{t+1} = b_0^{(q)}x_t + b_1^{(q)}x_{t-1} + \dots + b_q^{(q)}x_{t-q} = \sum_{i=0}^q b_i^{(q)}x_{t-i}$. The orders A and L of the ARDL model are determined in a similar manner. The methods presented above for determining the orders of AR, VAR, and ARDL models are used to implement the structural adaptation of the error predictors in online-mode, i.e., at each discrete time step.

The coefficients of the AR model are determined using the Yule-Walker equation:

$$\mathbf{W}_{YW} = P_\tau^{-1} \rho^{(\tau)}, \quad (18)$$

where $\mathbf{W}_{YW} = (w_1 \dots w_j \dots w_p)^T$ is the coefficients vector of AR model order p ; $\rho^{(\tau)} = (r_1 \dots r_j \dots r_\tau)^T$ is the vector of autocorrelation estimates before the lag $\tau = p$; P_τ is the Toeplitz matrix based on $\rho^{(\tau)}$ determined as in the (16). The tasks (8) and (10) for finding the parameters of the VAR and ARDL models are solved using the least squares method.

4. EXPERIMENTAL VALIDATION AND DISCUSSION OF RESULTS

The proposed approach for developing a multi-output soft sensor with a predictor was tested as follows. To implement the error predictors that account for the correlation of soft sensor errors for the fractional composition (FC), pairs are considered: the first pair is the initial boiling point of fractional composition (IBP FC) and 10% FC; the second pair is 50% FC and 95% FC. Figures 3 and 4 show the presence of a dynamic relationship between the selected pairs of soft sensor error time series using smoothed squared coherency spectral estimate smoothed sample estimates [30] of the squared coherence spectrum :

$$\bar{K}_{12}^2(\lambda) = \frac{\bar{A}_{12}^2(\lambda)}{\bar{C}_{11}(\lambda)\bar{C}_{22}(\lambda)}, \quad 0 \leq \lambda \leq F, \quad (19)$$

where $\bar{A}_{12}(\lambda) = \sqrt{\bar{L}_{12}^2(\lambda) + \bar{Q}_{12}^2(\lambda)}$, $0 < \lambda < F$ is the smoothed cross amplitude spectral estimate; $\bar{L}_{12}(\lambda) = 2 \left(\tilde{l}_{12}(0) + 2 \sum_{\kappa=1}^{L-1} \tilde{l}_{12}(\kappa) \beta(\kappa) \cos \frac{\pi \kappa \lambda}{F} \right)$, $0 \leq \lambda \leq F$ and $\bar{Q}_{12}(\lambda) = 4 \sum_{\kappa=1}^{L-1} \tilde{q}_{12}(\kappa) \beta(\kappa) \sin \frac{\pi \kappa \lambda}{F}$,

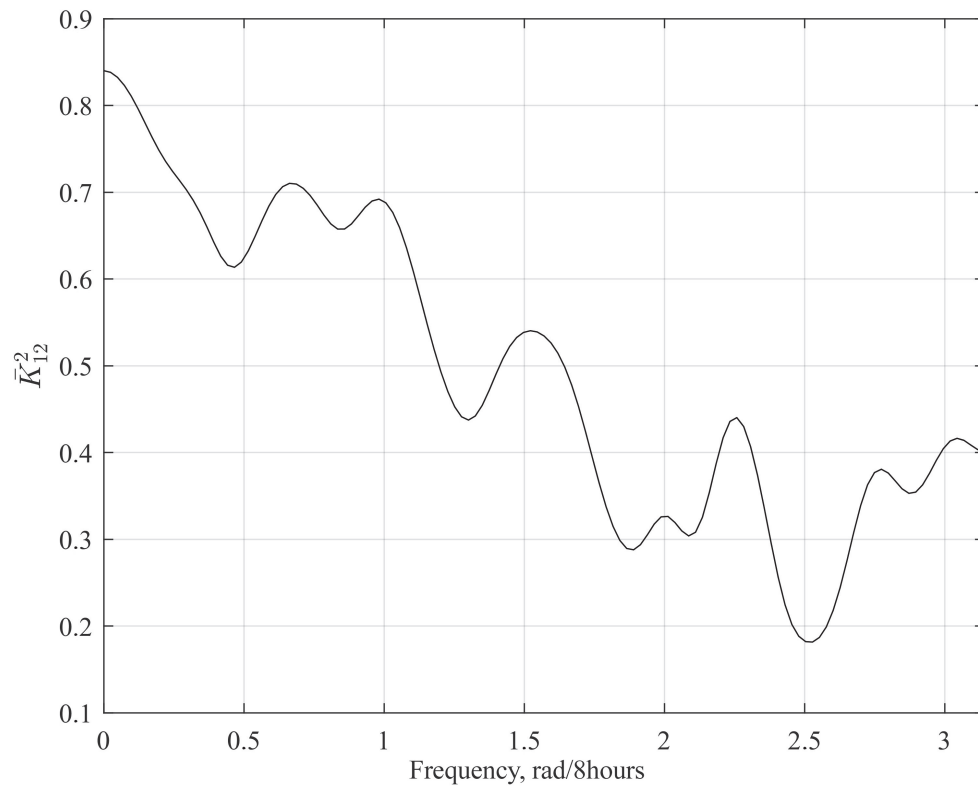


Fig. 3. Smoothed sample estimate of the quadrature coherency spectrum of SS IBP FC and FC 10% errors on the training sample.

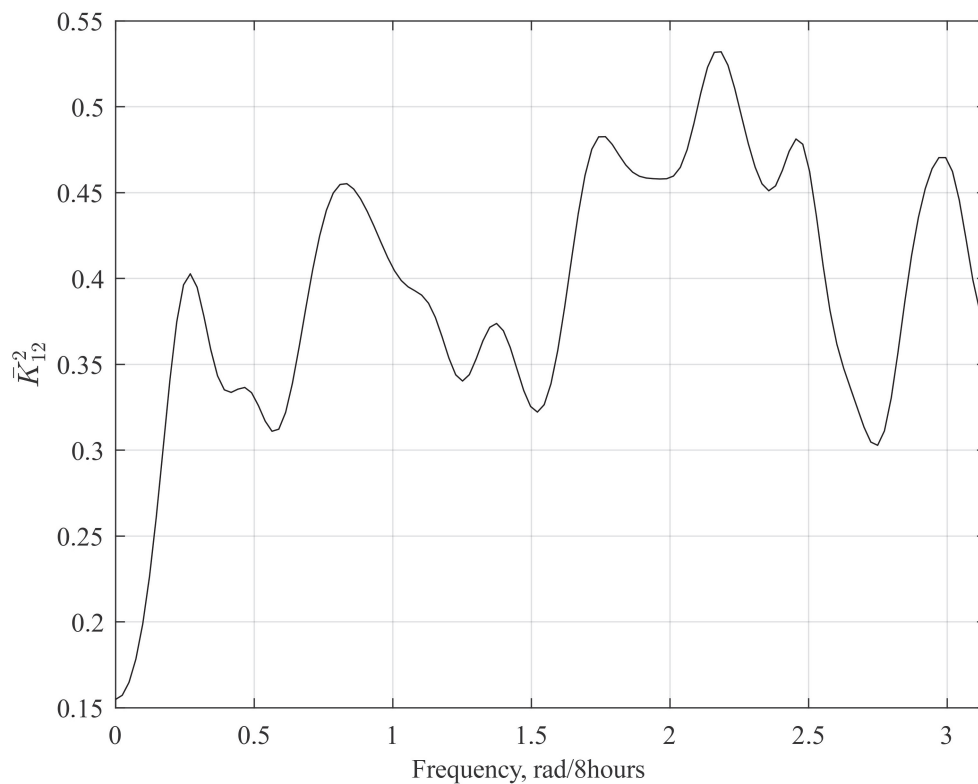


Fig. 4. Smoothed sample estimate of the quadrature coherency spectrum of SS FC 50% and FC 95% errors on the training sample.

$1 \leq \lambda \leq F - 1$ are the smoothed co- and quadrature spectral estimates, respectively; $\tilde{l}_{12}(\kappa) = 0.5(c_{12}(\kappa) + c_{12}(-\kappa))$, $0 \leq \kappa \leq L - 1$ are the even cross covariance function estimates; $\tilde{q}_{12}(\kappa) = 0.5(c_{12}(\kappa) - c_{12}(-\kappa))$, $0 \leq \kappa \leq L - 1$ are the odd cross covariance function estimates; $c_{12}(\kappa) = \frac{1}{n} \sum_{t=1}^{n-\kappa} (e_{1,t} - \bar{e}_1)(e_{2,t+\kappa} - \bar{e}_2)$, $c_{12}(-\kappa) = \frac{1}{n} \sum_{t=1}^{n-\kappa} (e_{1,t+\kappa} - \bar{e}_1)(e_{2,t} - \bar{e}_2)$, $0 \leq \kappa \leq L - 1$ is the cross covariance function estimate; $\bar{C}_{11}(\lambda) = 2 \left(c_{11}(0) + 2 \sum_{\kappa=1}^{L-1} c_{11}(\kappa) \beta(\kappa) \cos \frac{\pi \kappa \lambda}{F} \right)$, $0 \leq \lambda \leq F$ is the smoothed spectral estimate; $\beta(\kappa) = \begin{cases} 1 - \frac{|\kappa|}{M}, & |\kappa| \leq M \\ 0, & |\kappa| > M \end{cases}$ is the Bartlett window; $c_{11}(\kappa) = \frac{1}{n} \sum_{t=1}^{n-\kappa} (e_{1,t} - \bar{e}_1)(e_{1,t+\kappa} - \bar{e}_1)$, $0 \leq \kappa \leq L - 1$ is the autocovariance function estimate; $\bar{e}_1 = \frac{1}{n} \sum_{t=1}^n e_{1,t}$ is mean of the SS error; L is number of lags of used covariation functions; F is the maximum frequency of calculating spectral estimates; estimates $\bar{C}_{22}(\lambda)$ and $c_{22}(\kappa)$ are calculated similarly as $\bar{C}_{11}(\lambda)$ and $c_{11}(\kappa)$.

The amplitude values of the squared coherence spectrum shown in Fig. 3 are greater than 0.6 in the low-frequency region, indicating a close relationship between the soft sensor errors for IBP FC and 10% FC and low autocorrelation of the error series for these indicators. The graph in Fig. 4 reflects a relationship between the soft sensor errors for 50% FC and 95% FC, but there is significant autocorrelation of the soft sensor errors for one of the indicators in the low-frequency region, which is expressed by a low amplitude value of the smoothed sample estimate of the squared coherence spectrum, around 0.15.

The proposed MOSS is compared with an error predictor based on an AR model (which does not account for SS error correlation and is widely used in industry) through the procedure of bias update (BU) for the MLR model within the SS. The proposed approach is implemented for both adaptive (AdaptSS) and non-adaptive SS. The adaptation is implemented within the framework of the moving window (MW) method. Versions of the error predictors with structural adaptation (SA) and parametric adaptation (PA) are also implemented. PA is understood as the recalculation of the predictor parameters on the shifted MW without changing their order, which is determined by the methods described in the previous section during the first MW iteration. SA is understood as the calculation of the predictor parameters along with the determination of their order at each shift of the MW.

The total available sample contains 1382 observations. It is divided into a training sample with 967 observations and a test sample with 415 observations. For adaptive soft sensors and predictors, the size of the SW is the same as the size of the training sample for non-adaptive SS.

Tables 3 and 4 present the test results for non-adaptive and adaptive soft sensors, respectively. The coefficient of determination R_j^2 and the mean absolute error (MAE_j) were used as accuracy criteria for the j th quality indicator:

$$R_j^2 = 1 - \frac{\sum_{i=1}^n (y_{j,i} - \hat{y}_{j,i})^2}{\sum_{i=1}^n (y_{j,i} - \bar{y}_j)^2}, \quad (20)$$

$$\text{MAE}_j = \frac{1}{n} \sum_{i=1}^n |y_{j,i} - \hat{y}_{j,i}|, \quad (21)$$

where $y_{j,i}$ – laboratory value of j th quality indicator; $\hat{y}_{j,i}$ – estimate of j th quality indicator; $\bar{y}_j = \frac{1}{n} \sum_{i=1}^n y_{j,i}$.

It should be noted that the model parameters were found according to criterion (6). Criteria (20) and (21) are used for comparative analysis due to their widespread use in monitoring and automation systems for industrial processes.

Table 3. Testing results for non-adaptive SS

Type of SS	Pair with correlated errors				Pair with correlated errors			
	IBP FC		10% FC		50% FC		95% FC	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
Industrial approaches								
SS MLR	0.250	13.590	0.506	10.177	0.824	4.735	0.274	7.302
SS BU	0.667	9.023	0.776	6.808	0.905	3.293	0.601	5.105
The proposed method in comparison with the AR model (non-adaptive)								
SS AR	0.872	5.108	0.851	4.974	0.928	2.606	0.375	6.418
SS ARDL	0.868	5.180	0.909	3.731	0.927	2.679	0.707	4.047
SS VAR	0.868	5.184	0.909	3.732	0.928	2.678	0.707	4.048
SS AR ARDL	0.873	5.063	0.839	5.166	0.928	2.606	0.397	6.298
SS AR VAR	0.872	5.108	0.851	4.975	0.928	2.607	0.375	6.416
The proposed method in comparison with the AR model (parametric adaptation)								
SS AR PA	0.887	4.558	0.829	5.373	0.933	2.443	0.372	6.378
SS ARDL PA	0.883	4.654	0.914	3.510	0.933	2.473	0.715	3.910
SS VAR PA	0.883	4.647	0.914	3.507	0.933	2.472	0.715	3.909
SS AR PA ARDL PA	0.888	4.522	0.819	5.552	0.933	2.450	0.395	6.269
SS AR PA VAR PA	0.887	4.551	0.829	5.375	0.933	2.441	0.373	6.375
The proposed method in comparison with the AR model (structural adaptation)								
SS AR SA	0.885	4.652	0.828	5.366	0.933	2.479	0.372	6.378
SS ARDL SA	0.883	4.654	0.914	3.510	0.933	2.473	0.715	3.910
SS VAR SA	0.883	4.607	0.914	3.485	0.933	2.472	0.715	3.909
SS AR SA ARDL SA	0.887	4.565	0.817	5.592	0.932	2.469	0.395	6.274
SS AR SA VAR SA	0.885	4.645	0.828	5.369	0.933	2.478	0.373	6.375

Table 4. Testing results for adaptive SS

Type of SS	Pair with correlated errors				Pair with correlated errors			
	IBP FC		10% FC		50% FC		95% FC	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
Industrial approaches								
AdaptSS MLR	0.549	10.642	0.659	8.522	0.892	3.401	0.494	5.579
AdaptSS BU	0.771	7.504	0.828	5.899	0.930	2.566	0.677	4.251
The proposed method in comparison with the AR model (non-adaptive)								
AdaptSS AR	0.877	4.904	0.885	4.131	0.938	2.279	0.545	5.328
AdaptSS ARDL	0.877	4.935	0.912	3.694	0.939	2.268	0.729	3.657
AdaptSS VAR	0.877	4.938	0.912	3.694	0.939	2.267	0.729	3.659
AdaptSS AR ARDL	0.879	4.861	0.877	4.268	0.938	2.272	0.560	5.225
AdaptSS AR VAR	0.877	4.905	0.885	4.131	0.938	2.280	0.546	5.325
The proposed method in comparison with the AR model (parametric adaptation)								
AdaptSS AR PA	0.879	4.857	0.885	4.137	0.939	2.247	0.545	5.334
AdaptSS ARDL PA	0.879	4.886	0.913	3.671	0.939	2.248	0.728	3.661
AdaptSS VAR PA	0.879	4.883	0.913	3.669	0.939	2.248	0.728	3.660
AdaptSS AR PA ARDL PA	0.880	4.810	0.877	4.286	0.938	2.245	0.561	5.217
AdaptSS AR PA VAR PA	0.879	4.854	0.885	4.137	0.939	2.246	0.545	5.332
The proposed method in comparison with the AR model (structural adaptation)								
AdaptSS AR SA	0.876	4.932	0.885	4.175	0.938	2.247	0.545	5.330
AdaptSS ARDL SA	0.879	4.886	0.913	3.671	0.939	2.248	0.728	3.661
AdaptSS VAR SA	0.882	4.802	0.915	3.647	0.939	2.248	0.728	3.660
AdaptSS AR SA ARDL SA	0.878	4.849	0.876	4.347	0.939	2.249	0.561	5.219
AdaptSS AR SA VAR SA	0.876	4.929	0.885	4.176	0.939	2.246	0.545	5.329

Based on Tables 3 and 4, for IBP FC estimation, the best results in terms of accuracy criteria are achieved using the MOSS with a WCEP of type (2). For the 10% FC estimation, the MOSS with a CEP of type (1) shows the best results. For the 95% FC, the MOSS with a CEP of type (1) also shows the best results. A high degree of similarity is noted in the results of the MOSS with a CEP of type (1) based on VAR and ARDL models, and their significant advantage over the SS with an error predictor based on an AR model is also noted.

According to Fig. 4 and Tables 3, 4, the minor effect from using the CEP for the 50% FC SS (in comparison with the CEP for other SSs) is associated with an insufficiently strong dependence of the 50% FC SS errors on the 95% FC SS errors, as well as the presence of significant serial correlation in the 50% FC SS errors.

Tables 5 and 6 show the percentage change in accuracy criteria for the best result relative to the baseline methods (SS, SS BU, SS AR, SS AR PA, SS AR SA), which do not account for error cross-correlation for non-adaptive and adaptive SS, respectively. According to Tables 5 and 6, the lowest mean absolute error (MAE) is most often achieved using a CEP based on a VAR model with parametric or structural adaptation (VAR PA, VAR SA are highlighted in bold in the tables).

Table 5. Change of the accuracy criteria for non-adaptive SS

Type of SS	Pair with correlated errors				Pair with correlated errors			
	IBP FC		10% FC		50% FC		95% FC	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
Basic method								
SS MLR	71.83%	66.73%	44.62%	65.76%	11.67%	48.41%	61.69%	46.47%
SS BU	24.93%	49.88%	15.10%	48.81%	3.08%	25.81%	15.92%	23.44%
SS AR	1.82%	11.47%	6.95%	29.94%	0.53%	6.26%	47.55%	39.10%
SS AR PA	0.11%	0.79%	9.32%	35.15%	0.00%	0.00%	47.90%	38.72%
SS AR SA	0.36%	2.79%	9.39%	35.07%	0.08%	1.48%	47.90%	38.72%
Best result								
	SS AR PA ARDL PA		SS VAR SA		SS AR PA		SS VAR PA	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
	0.888	4.522	0.914	3.485	0.933	2.443	0.715	3.909

Table 6. Change of the accuracy criteria for adaptive SS

Type of SS	Pair with correlated errors				Pair with correlated errors			
	IBP FC		10% FC		50% FC		95% FC	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
Basic method								
AdaptSS MLR	37.71%	54.88%	27.97%	57.20%	4.96%	33.99%	32.14%	34.39%
AdaptSS BU	12.53%	36.00%	9.43%	38.17%	0.92%	12.53%	6.99%	13.90%
AdaptSS AR	0.53%	2.08%	3.30%	11.70%	0.03%	1.51%	25.04%	31.30%
AdaptSS AR PA	0.34%	1.12%	3.25%	11.84%	-0.04%	0.10%	25.12%	31.38%
AdaptSS AR SA	0.61%	2.64%	3.21%	12.65%	-0.01%	0.10%	25.09%	31.33%
Best result								
	AdaptSS VAR SA		AdaptSS VAR SA		AdaptSS AR PA ARDL PA		AdaptSS VAR PA	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
	0.882	4.802	0.915	3.647	0.938	2.245	0.728	3.660

The results presented in Tables 3–6 make it possible to conclude that the proposed method (SS/AdaptSS VAR, ARDL, AR VAR, AR ARDL) is effective compared to the existing methods considered. For non-adaptive and adaptive SS, the reduction in MAE compared to BU was on average 37 and 25.1%, respectively. For FC IBP – 49.9 and 36%, for FC 10% – 48.8 and 38.2%, for FC 50% – 25.8 and 12.5%, for FC 90% – 23.4 and 13.9%. Compared to the AR-based error predictor, the MAE reduction for FC IBP was 11.5 and 2.1%, for FC 10% – 29.9 and 11.7%, for FC 50% – 6.3 and 1.5%, for FC 90% – 39.1 and 31.3%, averaging 21.7 and 11.7% for non-adaptive and adaptive SS, respectively.

5. CONCLUSION

The problem of developing a MOSS with a correlated error predictor has been solved. The effectiveness of the proposed approach for developing a MOSS with a predictor has been demonstrated for use in a quality control system for the fractional composition indicators of the light diesel fraction from a complex fractionating column in a hydrocracking unit. The comparison of the proposed MOSS with the BU and the AR model-based error predictor without considering error cross-correlation showed a reduction in MAE by an average of 29.3 and 21% and an increase in the coefficient of determination by 14 and 7% for non-adaptive and adaptive soft sensors, respectively.

FUNDING

The research was carried out within the state assignment of IACP FEB RAS on theme FW-2025-0002 (implementation of the proposed method and testing on experimental data) and theme FWWF-2021-0003 (method for constructing a multidimensional SS with an error predictor).

REFERENCES

1. Lawrence, N.P., Damarla, S.K., Kim, J.W., Tulsyan, A., Amjad, F., Wang, K., Chachuat, B., Lee, J.M., Huang, B., and Gopaluni, R.B., Machine Learning for Industrial Sensing and Control: A Survey and Practical Perspective, *Control Engineering Practice*, 2024, vol. 145, 105841. <https://doi.org/10.1016/j.conengprac.2024.105841>
2. Bakhtadze, N.N., Virtual Analyzers: Identification Approach, *Autom. Remote Control*, 2004, vol. 65, no. 11, pp. 1691–1709. <https://doi.org/10.1023/B:AURC.0000047885.52816.c7>
3. Logunov, P.L., Shamanin, M.V., Kneller, D.V., Setin, S.P., and Shunderiyuk, M.M., Advanced Process Control: From a PID loop up to refinery-wide optimization, *Autom. Remote Control*, 2020, vol. 81, pp. 1929–1943. <https://doi.org/10.1134/S0005117920100100>
4. Shvarts, V.I., Determination of Rectification Column Characteristics on the Basis of Normal Operation Data by Potential Function Method, *Autom. Remote Control*, 1968, no. 6, pp. 142–144.
5. Mejdell, T. and Skogestad, S., Estimation of Distillation Compositions from Multiple Temperature Measurements Using Partial-Least-Squares Regression, *Industrial & Engineering Chemistry Research*, 1991, vol. 30, no. 12, pp. 2543–2555. <https://doi.org/10.1021/ie00060a007>
6. Torgashov, A. and Skogestad, S., The Use of First Principles Model for Evaluation of Adaptive Soft Sensor for Multicomponent Distillation Unit, *Chemical Engineering Research and Design*, 2019, vol. 151, pp. 70–78. <https://doi.org/10.1016/j.chemd.2019.08.017>
7. Ferreira, J., Pedemonte, M., and Torres, A.I., Development of a Machine Learning-Based Soft Sensor for an Oil Refinery's Distillation Column, *Computers & Chemical Engineering*, 2022, vol. 161, 107756. <https://doi.org/10.1016/j.compchemeng.2022.107756>
8. Cheresenko, A.A., Soft Sensors Based on Digital Models, *Autom. Remote Control*, 2023, vol. 84, no. 7, pp. 788–796. <https://doi.org/10.1134/s0005117923070044>

9. Wang, K., Shang, C., and Liu, L., Dynamic Soft Sensor Development Based on Convolutional Neural Networks, *Industrial & Engineering Chemistry Research*, 2019, vol. 58, no. 26, pp. 11521–11531. <https://doi.org/10.1021/acs.iecr.9b02513>
10. Xibilia, M.G., Latino, M., and Marinkovic, Z., Soft Sensors Based on Deep Neural Networks for Applications in Security and Safety, *IEEE Transactions on Instrumentation and Measurement*, 2020, vol. 69, no. 10, pp. 7869–7876. <https://doi.org/10.1109/TIM.2020.2984465>
11. Xuefeng, Y., Hybrid Artificial Neural Network Based on BP-PLSR and Its Application in Development of Soft Sensors, *Chemometrics and Intelligent Laboratory Systems*, 2010, vol. 103, no. 2, pp. 152–159. <https://doi.org/10.1016/j.chemolab.2010.07.002>
12. Li, L., Li, N., and Wang, X., Multi-Output Soft Sensor Modeling Approach for Penicillin Fermentation Process Based on Features of Big Data, *Expert Systems with Applications*, 2023, vol. 213, pp. 119208. <https://doi.org/10.1016/j.eswa.2022.119208>
13. Mastelini, S.M., Da Costa, V.G.T., and Santana, E.J., Multi-Output Tree Chaining: An Interpretative Modelling and Lightweight Multi-Target Approach, *J. Signal Proc. Syst.*, 2019, vol. 91, no. 2, pp. 191–215. <https://doi.org/10.1007/s11265-018-1376-5>
14. Xu, S., An, X., and Qiao, X., Multi-Output Least-Squares Support Vector Regression Machines, *Pattern Recognition Letters*, 2013, vol. 34, no. 9, pp. 1078–1084. <https://doi.org/10.1016/j.patrec.2013.01.015>
15. Grbić, R., Slisković, D., and Kadlec, P., Adaptive Soft Sensor for Online Prediction and Process Monitoring Based on a Mixture of Gaussian Process Models, *Computers & Chemical Engineering*, 2013, vol. 58, pp. 84–97. <https://doi.org/10.1016/j.compchemeng.2013.06.014>
16. Bakirov, R., Gabrys, B., and Fay, D., Multiple Adaptive Mechanisms for Data-Driven Soft Sensors, *Computers & Chemical Engineering*, 2017, vol. 96, pp. 42–54. <https://doi.org/10.1016/j.compchemeng.2016.08.017>
17. Saptoro, A., State of the Art in the Development of Adaptive Soft Sensors Based on Just-in-Time Models, *Procedia Chemistry*, 2014, vol. 9, pp. 226–234. <https://doi.org/10.1016/j.proche.2014.05.027>
18. Pan, B., Jin, H., Wang, L., Qian, B., Chen, X., Huang, S., and Li, J. Just-in-Time Learning Based Soft Sensor with Variable Selection and Weighting Optimized by Evolutionary Optimization for Quality Prediction of Nonlinear Processes, *Chemical Engineering Research and Design*, 2019, vol. 144, pp. 285–299. <https://doi.org/10.1016/j.cherd.2019.02.004>
19. Fujiwara, K., Kano, M., Hasebe, S., and Takinami, A., Soft-Sensor Development Using Correlation-Based Just-in-Time Modeling, *AIChE J.*, 2009, vol. 55, no. 7, pp. 1754–1765.
20. Sangoi, E., Sanseverinatti, C.I., Clementi, L.A., and Vega, J.R., A Bayesian Bias Updating Procedure for Automatic Adaptation of Soft Sensors, *Computers & Chemical Engineering*, 2021, vol. 147, pp. 107250. <https://doi.org/10.1016/j.compchemeng.2021.107250>
21. Xiong, W., Zhang, W., Xu, B., and Huang, B., JITL Based MWGPR Soft Sensor for Multi-Mode Process with Dual-Updating Strategy, *Computers & Chemical Engineering*, 2016, vol. 90, pp. 260–267. <https://doi.org/10.1016/j.compchemeng.2016.04.033>
22. Klimchenko, V., Torgashov, A., Shardt, Y.A.W., and Yang, F., Multi-Output Soft Sensor with a Multivariate Filter That Predicts Errors Applied to an Industrial Reactive Distillation Process, *Mathematics*, 2021, vol. 9, no. 16, p. 1947. <https://doi.org/10.3390/math9161947>
23. Klimchenko, V.V., Snegirev, O.Yu., Shevlyagina, S.A., and Torgashov, A.Yu., Developing an Adaptive Soft Sensor Using a Predictive Filter for a Nonstationary Process, *Autom. Remote Control*, 2022, vol. 83, pp. 1984–1994. <https://doi.org/10.1134/S00051179220120104>
24. Torgashov, A., Snegirev, O., Klimchenko, V., and Yang, F., Development of a Multivariate Quality Estimator for an Industrial Fractionator in the Presence of Cross-Correlation of Output Variables, *IFAC-PapersOnLine*, 2023, vol. 56, no. 2, pp. 7160–7165. <https://doi.org/10.1016/j.ifacol.2023.10.596>
25. Hamilton, J.D., *Time Series Analysis*, Princeton: Princeton University Press, 1994.

26. Pesaran, M.H., *An Autoregressive Distributed Lag Modelling Approach to Cointegration Analysis*, Department of Applied Economics, University of Cambridge, Cambridge, 1995, vol. 9514, pp. 371–413.
27. Yeo, W.S., Saptoro, A., Kumar, P., and Kano, M., Just-in-Time Based Soft Sensors for Process Industries: A Status Report and Recommendations, *J. Process Control*, 2023, vol. 128, 103025.
<https://doi.org/10.1016/j.jprocont.2023.103025>
28. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., and Ljung G.M., *Time series analysis: forecasting and control fifth ed.*, Hoboken, New Jersey: Wiley, 2015.
29. Alt, F.L., Distributed Lags, *Econometrica*, 1942, vol. 10, no. 2, pp. 113–128.
<https://doi.org/10.2307/1905784>
30. Jenkins, G.M. and Watts, D.G., *Spectral Analysis and Its Applications*, San Francisco: Holden-Day, 1968.

This paper was recommended for publication by N.N. Bakhtadze, a member of the Editorial Board

A Control Algorithm for Player's Actions in a Triopoly Game under Linear Demand and Cost Functions

M. I. Geraskin

Samara University, Samara, Russia

e-mail: innovation@ssau.ru

Received November 29, 2024

Revised March 21, 2025

Accepted March 25, 2025

Abstract—This paper considers a game of three oligopolistic firms (a triopoly) with the linear demand and cost functions of players. The reflexive behavior of players is investigated through a formalization with conjectural variations, i.e., players' expectations regarding the impact of their actions on the counterparty's action. A method is developed to calculate the optimal (consistent) sum of conjectural variations of a given player in terms of the utility functions of the other players (environment). An algorithm is proposed to control the player's actions by the environment: implementing this algorithm forms a given mental type of the player and predetermines the latter's goal-oriented behavior. Computer simulations are carried out for a hierarchical game in which the environment (the Principal) applies informational control to the player's actions, using the Russian telecommunications market as an example. According to the simulation results, the control algorithm is effective and increases the environment's utility.

Keywords: oligopoly, conjectural variation, hierarchical game, optimal control, telecommunications market

DOI: 10.31857/S0005117925070043

1. INTRODUCTION

The game-theoretic model of oligopoly as an economic sphere in which several firms sell an identical product formalizes the interaction of these firms as players. As a rule, players are treated as price takers: they face the equilibrium market price on the aggregate demand curve and perform game actions (choose their strategies) in the form of supply quantities of the product. The key feature of this game is that the equilibrium price depends on the actions of all players, so when choosing a strategy, players must predict the actions of their counterparties (the environment). The change in the environment's action expected by a player in response to a unit change in the latter's action is called a conjectural variation. The player's a priori unawareness regarding the conjectural variations of the environment predetermines the fundamental complexity of the oligopoly game. As is known from the classical models of A. Cournot [1] and H. Stackelberg [2], the optimal strategies of players depend on two factors: the inverse demand function for the product and the cost functions of the players. Therefore, the simpler form these functions have, the more complete and informative the analysis of oligopoly game outcomes will be.

The oligopoly game with the linear demand function for a product and linear cost functions of players is the most typical case considered by researchers [3–17], since this model is the most convenient tool for studying game strategies. In particular, dynamic oligopoly games based on finite-difference equations of players' responses [3–6] and dynamic games with differential equations describing the change process of the utility functions of players [7–9] were studied in such a formulation. A dynamic game based on fractal differential equations of changes in the players' actions was investigated as well [10]. Also, static oligopoly games were introduced and Cournot and Stackelberg equilibria were compared accordingly [11].

Applied aspects of the oligopoly game with linear demand and cost functions were considered in the context of choosing a firm's tax policy [12], assessing the consequences of merging firms [13], entering a bilateral oligopoly game by a new firm [14], selling product sets [15, 16], and adopting new technology [17].

The above studies involved a non-hierarchical game-theoretic model of oligopoly with initially equal players, even in the case of Stackelberg leadership: being not conditioned by an original hierarchy of players, leadership arises from their asymmetric awareness in game dynamics. The awareness levels were formalized using reflexion ranks r , and reflexion was understood as the process of making conjectures by a player regarding the strategies of other players (the environment) [18]. As a result of this process, each player creates a set of phantom environment players in the mind, which can be followers at reflexion rank 1, first-level Stackelberg leaders at reflexion rank 2, and $(r - 1)$ th-level Stackelberg leaders at reflexion rank r . In other words, at each reflexion rank, the player thinks that the environment consists of players of a lower reflexion rank. Relating these theoretical considerations to business practice, we emphasize that the decision-maker (DM) representing a firm may have no idea of Stackelberg leadership as a scientific category. Nevertheless, when choosing an appropriate action in the market, the DM must analyze the possible responses of the counterparties (i.e., must perform reflexion). Therefore, if a player correctly predicts the environment's beliefs (i.e., the phantoms coincide with the real players), he/she will dominate in equilibrium in accordance with the leadership level [19]. But in a non-hierarchical model of the game, the leader does not actually control the environment players.

However, in reality, it is easy to imagine a game situation where a certain group of players can choose a common strategy aimed at increasing their payoffs at the expense of one player. Therefore, a topical issue is to study the following hierarchical system of oligopoly players: a group of players (the Principal) controls the strategy of a certain player by choosing a group strategy, inducing this player to choose an optimal strategy in terms of the group's utility functions. In this case, the control process is also based on the above reflexive behavior of players. Below, this problem is investigated for the three-player game model with linear demand and cost functions, which makes the analysis of equilibria explicit.

2. THE NON-HIERARCHICAL GAME WITH LINEAR DEMAND AND COST FUNCTIONS (THE LINEAR MODEL)

The non-hierarchical model of an oligopoly market describes the behavior of n players (firms) supplying an identical product to the market. The consumer demand for this product is characterized by an inverse demand function $P(Q)$, decreasing with the total sales quantity Q ($P'_Q < 0$). The players are equal and choose actions (strategies) in the form of their supply quantities Q_i , in accordance with their increasing cost functions $C_i(Q_i)$ ($C''_{Q_i} > 0$), with the goal of maximizing the utility function $\pi_i(Q, Q_i) = P(Q)Q_i - C_i(Q_i)$.

Consider the oligopoly game in the case of the linear demand and cost functions of players:

$$\begin{aligned} P(Q) &= a - bQ, \quad A > 0, \quad b > 0, \quad a \gg b, \\ C_i(Q_i) &= B_{0i} + B_i Q_i, \quad B_{0i} \geq 0, \quad B_i > 0, \end{aligned}$$

where Q_i denotes the action of player i ; a, b, B_{0i} , and B_i are constant coefficients expressing the parameters of the demand function (the maximum price a and the price decline rate b) and the parameters of the cost function (the fixed costs B_{0i} and the marginal costs B_i). The total action of the players is $Q = \sum_{i \in N} Q_i$.

In this case, the players seek to maximize their utility functions π_i as follows:

$$\max_{Q_i \geq 0} \pi_i(Q, Q_i) = \max_{Q_i \geq 0} [(a - bQ)Q_i - B_{0i} - B_i Q_i], \quad i \in N = \{1, \dots, n\}, \quad (1)$$

where N stands for the set of players and n is the number of players.

The Nash equilibrium Q_i^* in the non-hierarchical game $\Gamma = \langle N, \{Q_i, i \in N\}, \{\pi_i, i \in N\} \rangle$ is calculated based on the necessary optimality conditions $\frac{\partial \pi_i(Q_i^*, \rho_{ij})}{\partial Q_i} = 0$, $i, j \in N$, which involve the conjectural variation of player i . For the system of the objective functions (1), the necessary optimality conditions have the form

$$a - bQ - b(1 + S_i^r)Q_i - B_i = 0, \quad i \in N, \quad S_i^r = \sum_{j \in N \setminus i} \rho_{ij}^r, \quad (2)$$

where S_i^r is the sum of conjectural variations of player i at reflexion rank r . Due to equations (2), the sum of conjectural variations (SCV) of each player is crucial for calculating the game equilibrium: the other parameters in (2) can be considered common knowledge, and the player's SCV value may be unknown to the environment.

With the player's type parameter denoted by $\alpha_i = \frac{a - B_i}{b}$, system (2) can be conveniently written as

$$\alpha_i - (2 + S_i^r)Q_i - Q_j \sum = 0, \quad i \in N, \quad (2a)$$

where $Q_j \sum$ is the total action of the environment of player i ; the environment is designated as the generalized player j .

A comparative analysis of (1) and (2a) allows deriving the maximum profit of player i depending on the latter's equilibrium action, i.e., the function $\pi_i^* = \pi_i(Q_i^*)$, in the case of reflexive behavior of players.

Proposition 1. *Under the reflexive behavior of all players, the dependence of the maximum of the player's utility function on his/her equilibrium action has the form*

$$\pi_i^* = b(1 + S_i^r)(Q_i^*)^2 - B_{0i}. \quad (3)$$

Proof. From (1) it follows that

$$\pi_i = \left(a - b(Q_i + Q_j \sum) \right) - B_i Q_i - B_{0i} = b \left(\alpha_i - (Q_i + Q_j \sum) \right) Q_i - B_{0i};$$

on the other hand, by formula (2a),

$$\alpha_i - Q_i^* - Q_j^* \sum = (1 + S_i^r)Q_i^*.$$

Substituting this result into the former expression finally gives (3).

According to equations (2), the solution of this system (the vector of equilibrium actions $\mathbf{Q}^* = \{Q_i^*, i \in N\}$) depends on the SCV vector $\mathbf{S} = \{S_i, i \in N\}$. Therefore, if the SCV value of some player i is such that the utility functions of the other players (the environment) are maximized in equilibrium, then we can say that the environment purposefully controls this player (i.e., a hierarchical game arises).

3. THE HIERARCHICAL GAME IN THE LINEAR TRIOPOLY MODEL

Consider an oligopoly game with three players (i.e., the triopoly case). In this model, one player i opposes all other players (the environment), which have a common goal of reaching a beneficial action of the former player from their standpoint. Therefore, player i will be called the controlled player. Let the number j be associated with all environment players taken together.

We describe the awareness system in the hierarchical game with the reflexive behavior of players by the following assumptions.

1) All players are completely aware of each other's utility functions and actions and choose their actions simultaneously; the controlled player and the environment are independent of each other.

2) Environment players j are informed about each other's conjectural variations and are not informed about the SCV values of the controlled player; they choose the same strategies (actions denoted by q) and have the same SCV value (denoted by s). In addition, suppose that the environment players are identical in the type parameter α_j , $j \in N \setminus i$, with α being the average value of this parameter. Accordingly, the values of the environment's objective functions will be the same, denoted by π . The environment players can change the SCV value in a coordinated manner depending on the SCV value of the controlled player. Moreover, this dependence is inverse: increasing the SCV value of the controlled player (lowering his/her leadership level) reduces the SCV value of the environment (i.e., raises its leadership level). We formalize this assumption as follows:

$$\begin{aligned} q &= Q_j, \quad \alpha = \frac{1}{n-1} \sum_{j \in N \setminus i} \alpha_j, \quad \pi = \pi_j, \\ s &= S_j, \quad s' = \frac{\partial S_j}{\partial S_i} < 0, \quad s' = \text{const} \quad \forall j \in N \setminus i. \end{aligned} \quad (4)$$

3) At each time instant (step) of the game, the controlled player i chooses an action independently of the environment's actions by considering the latter's previous actions. The controlled player is unaware of the SCV values of the environment but can indirectly estimate them by restoring the optimal reactions (best responses) from the observed actions of the environment and his/her actions. The controlled player determines the SCV value as a Stackelberg leader (i.e., by differentiating the optimal reaction function of the environment).

4) Both counterparties (the controlled player and the environment) are unaware of each other's true leadership levels: they do not know the opponent's SCV value in the game.

The last assumption is crucial for organizing an informational control process by manipulating the controlled player's actions. With a goal-oriented action, the environment can create a controlled player's belief that the former's SCV value has changed appreciably, although the true action may be unit, leaving the environment's reaction curve and SCV value actually unchanged.

Based on these assumptions, we formulate a hierarchical model of the game: if the controlled player chooses an optimal action considering the available data on the environment's SCV values, then the environment maximizes its utility functions, inducing the controlled player to choose the best SCV value from the former's standpoint. In other words, the environment solves the following problem:

$$\max_{S_i} \pi^*(q(s, S_i), Q_i(s, S_i)), \quad s = s^0 = \text{const}, \quad (5)$$

provided that the controlled player chooses the strategy from the condition

$$\max_{Q_i} \pi_i(Q(s, S_i), q(s, S_i)), \quad (6)$$

where s^0 is the environment's SCV value at the initial step of the game.

Consider the environment's control process for the behavior of player i . For this purpose, we determine the optimal SCV value $\bar{S}_i$ of this player in terms of the environment's utility function (5): $\bar{S}_i = \arg \max_{S_i} \pi(q(s, S_i), Q_i(s, S_i))$.

Proposition 2. *The controlled player's SCV value maximizing the utility function (5) of the environment is calculated by solving the equation*

$$2(1+s)q_{S_i}^* + q^*s' = 0 \quad (7)$$

under the condition

$$s' < -\frac{(1+s)q^*}{2q_{S_i}^{*'}}. \quad (7a)$$

Proof. By the first-order optimality condition applied to the environment's objective function π^* (3), we have

$$\pi_{S_i}^{*'} = 2b(1+s)q_{S_i}^{*'}q^* + b(q^*)^2s' = 0,$$

which immediately gives (7). Let a change in the SCV value weakly impact the shift of the equilibrium $q_{S_i}^{*''}$. In view of $s'' = 0$ (see (4)), we transform the second-order optimality condition

$$\pi_{S_i S_i}^{*''} = b\{2s'q_{S_i}^{*'}q^* + 2(1+s)(q_{S_i}^{*''}q^* + q^{*2}) + 2q_{S_i}^{*'}q^*s' + q^{*2}s''\} < 0 \quad (7b)$$

to

$$2s'q_{S_i}^{*'} + (1+s)q^* < 0. \quad (7c)$$

The analysis of the Stackelberg leadership levels dynamics shows that $q_{S_i}^{*'} > 0$; in addition, $1+s > 0$ in the linear oligopoly model. Therefore, according to (7b), the maximum is achieved at $s' < 0$ (see Assumption (4)), particularly under condition (7a).

Note that equation (7) corresponds to the previously known formula [20] $2(1+S_i^r)Q_j^*Q_{jS_i}^{*'}P_Q' + ((1+S_i^r)P_{QS_i}^{*''} + P_Q' \frac{\partial S_i^r}{\partial S_i})Q_j^{*2} = 0$ for the general oligopoly problem, since $P_Q' = -b$ and $P_{QS_i}^{*''} = 0$ in the model with linear demand and cost functions under consideration.

4. OPTIMAL CONTROL METHODS

Equation (7) has been written for each environment player. The desired value $\bar{S}_i$ in (7) figures in the expressions for $q_{S_i}^{*'}, q^*$, which are necessary to calculate this unknown. Therefore, we consider a special case of a triopoly, where, without loss of generality, the second player will be the controlled one ($i = 2$) whereas the first and third players the environment ($j = 1, 3$).

Let us find the equilibrium in this game and the solution of the control problem in the hierarchical system (5).

Proposition 3. *In the triopoly game with players $j = 1, 3$ as the environment and player $i = 2$ as the controlled one, we have:*

i) *The optimal reaction functions of the controlled player and the environment are given by*

$$q = \frac{\alpha - Q_2}{3+s}, \quad Q_2 = \frac{\alpha_2 - 2q}{2+S_2}. \quad (8a)$$

ii) *The equilibrium actions have the form*

$$q^* = \frac{\alpha_2 - \alpha(2+S_2)}{2 - (2+S_2)(3+s)}, \quad Q_2^* = \frac{2\alpha - \alpha_2(3+s)}{2 - (2+S_2)(3+s)}. \quad (8b)$$

iii) *The controlled player's SCV value maximizing the utility function of the environment is calculated by solving the equation*

$$2(1+s)\frac{zq^* - \alpha}{y} + q^*s' = 0 \quad (8c)$$

under the condition

$$\frac{\pi_{S_i S_i}^{*''}}{b} = \frac{4q^*}{y}s' \left(\frac{zq^*}{2} + (1+s)q^* - \alpha \right) + 2(1+s)q^{*2} < 0, \quad (8d)$$

where $y = 2 - (2+S_2)(3+s)$ and $z = s'(2+S_2) + 3+s$.

Proof. In the case under consideration, system (2a) becomes

$$\begin{cases} \alpha_1 - (2 + S_1)Q_1 - Q_3 - Q_2 = 0 \\ \alpha_2 - (2 + S_2)Q_2 - Q_1 - Q_3 = 0 \\ \alpha_3 - (2 + S_3)Q_3 - Q_1 - Q_2 = 0, \end{cases} \Rightarrow \begin{cases} 2\alpha - 2(2 + s)q - 2Q_2 - 2q = 0 \\ \alpha_2 - (2 + S_2)Q_2 - 2q = 0. \end{cases}$$

Consequently, the optimal reaction functions of the second player and the environment take the form (8a). Resolving them yields the equilibrium actions (8b). For the sake of simplification, let us introduce $y = 2 - (2 + S_2)(3 + s)$ and $z = s'(2 + S_2) + 3 + s$. In this case, the derivative q_{S_2}' is calculated as

$$q_{S_2}' = \frac{\alpha_2 z - \alpha[y + z(2 + S_2)]}{y^2} = \frac{z[\alpha_2 - \alpha(2 + S_2)]}{y^2} - \frac{\alpha y}{y^2} = \frac{zq^* - \alpha}{y}.$$

Therefore, equation (7) turns into (8c).

Now we derive the second-order optimality condition (7a) for the environment's objective function. In view of $s'' = 0$, the value $q_{S_2 S_2}''$ can be expressed in the form

$$q_{S_2 S_2}'' = \frac{(z'q^* + zq_{S_2}')y - y'(zq^* - \alpha)}{y^2} = \frac{z'q^* + 2zq_{S_2}'}{y}$$

(more precisely than (7a)).

Substituting this expression into (7b), after appropriate transformations with $z' = 2s'$, we obtain

$$\begin{aligned} \frac{\pi_{S_i S_i}''}{b} &= q^* \left(4s' \frac{zq^* - \alpha}{y} + 2(1 + s) \frac{2s'q^* + 2zq_{S_2}'}{y} \right) + 2(1 + s)q^{*2} \\ &= \frac{4q^*}{y} \left(z[s'q^* + (1 + s)q_{S_2}'] + s'[(1 + s)q^* - \alpha] \right) + 2(1 + s)q^{*2}. \end{aligned}$$

According to (7), the change $(1 + s)q_{S_2}' = -\frac{s'q^*}{2}$ in the last equality finally leads to the sufficient maximum condition (8d).

Proposition 3 can be used to find $\overline{S}_2$.

Now we develop an algorithm for reaching the target value $\overline{S}_2$ from the environment's standpoint. Let an equilibrium (q^0, Q_2^0) be established at some initial step 0 of the game under some SCV values S_2^0 (the controlled player) and s^0 (the environment), which are unknown to the counterparties. Let us determine the environment's phantom reaction under which the controlled player will choose the SCV value $\overline{S}_2$. We denote by s_f the SCV value that the environment must have for the controlled player to determine his/her SCV value as $\overline{S}_2$. It will be called *the phantom SCV value of the environment*.

Proposition 4. *In the triopoly game, the controlled player $i = 2$ establishes the response according to $\overline{S}_2$ under the phantom SCV of the environment ($j = 1, 3$) if and only if*

$$s_f = -\frac{1}{\overline{S}_2} - 3. \quad (9)$$

Proof. If the controlled player performs the action Q_2 and the environment responds with the action q , the former will calculate the SCV value by differentiating the environment's reaction function. The environment's phantom reaction $q_f(Q_2)$ is calculated by substituting the unknown value s_f into (8a): $q_f = \frac{\alpha - Q_2}{3 + s_f}$. Then $S_2 = q_{Q_2}' = \left(\frac{\alpha - Q_2}{3 + s_f} \right)'_{Q_2} = -\frac{1}{3 + s_f}$. Therefore, if the environment's goal is to induce the controlled player's choice $\overline{S}_2$, then from $\overline{S}_2 = -\frac{1}{3 + s_f}$ we obtain the environment's SCV value (9) required.

Based on Proposition 4, we organize the following iterative control process with time instants (steps) $0, t, t+1, t+2, \dots$, indicated by the superscripts of the players' actions ($t > 0$).

1. At step t , the environment performs an action calculated according to its phantom reaction function:

$$q^t = \frac{\alpha - Q_2^0}{3 + s_f} = -\bar{S}_2(\alpha - Q_2^0).$$

2. At step $t+1$, the controlled player calculates the SCV value (9) given this action, which is equal to $\bar{S}_2$, and then responds according to the new reaction function $Q_2^{t+1} = \frac{\alpha_2 - 2q}{2 + \bar{S}_2}$:

$$Q_2^{t+1} = \frac{\alpha_2 - 2q^t}{2 + \bar{S}_2}.$$

3. At step $t+2$, the environment performs an action according to its true reaction function:

$$q^{t+2} = \frac{\alpha_2 - Q_2^{t+1}}{3 + s^0}.$$

4. At step $t+3$, the controlled player calculates the SCV value (9) given this action, which is equal to S_2^0 , and then returns to the original reaction function $Q_2^{t+3} = \frac{\alpha_2 - 2q}{2 + S_2^0}$.

5. At steps $t+4, t+5, \dots$, the environment and the controlled player repeat Stages 3 and 4, and the game equilibrium returns to the initial state (q^0, Q_2^0) .

At steps $t+1$ and $t+2$ of this process, the environment achieves the desired result and receives the maximum profit. As is known [3–6], the above iterative process converges to the equilibrium (8b) fast enough; therefore, suppose that the game state has already stabilized starting from step $t+3$. Consequently, the environment receives an additional payoff at steps $t+1$ and $t+2$ but may incur losses at step t . The payoffs at different steps must be brought to a comparable level based on discounting $\pi_d(t) = \pi^t e^{-\rho t}$, commonly used in dynamic oligopoly models [9]. To assess the effectiveness of the iterative control process, we compare the maximum values of the environment's utility function at steps 0, t , $t+1$, and $t+2$: control is effective if

$$\pi^{*t} e^{-\rho t} + \pi^{*t+1} e^{-\rho(t+1)} + \pi^{*t+2} e^{-\rho(t+2)} \geq \pi^{*0} (e^{-\rho t} + e^{-\rho(t+1)} + e^{-\rho(t+2)}),$$

where ρ denotes a discounting factor.

5. EXPERIMENTAL ANALYSIS OF OPTIMAL CONTROL

Consider a numerical experiment for the optimal control process. Computer simulations were carried out on the 2016–2021 demand and operator cost data for the Russian telecommunications market [22]. The linear demand function for the voice traffic of mobile operators was constructed in the form

$$P(Q) = a - bQ, \quad a = 1.6, \quad b = 0.000001;$$

the cost functions of mobile operators were taken without fixed costs due to the latter's negligible impact on the equilibrium:

$$C_i(Q_i) = B_i Q_i, \quad B_1 = 0.0005, \quad B_2 = 0.0018, \quad B_3 = 0.0004,$$

where $i = 1$ corresponds to MTS, $i = 2$ to MegaFon, and $i = 3$ to VimpelCom.

As in Section 4, the second player is the controlled one, with the action denoted by Q_2 , and the first and third players represent the environment ($j = 1, 3$), with the same actions denoted by q . Figure 1 shows the optimal reaction lines of the players (indicated by R) in the cases of Cournot equilibrium (the SCV values are $s = S_2 = 0$) and three-party Stackelberg leadership (the SCV values are $s = S_2 = -0.5$); the latter case is treated as the initial state of the game (point 0).

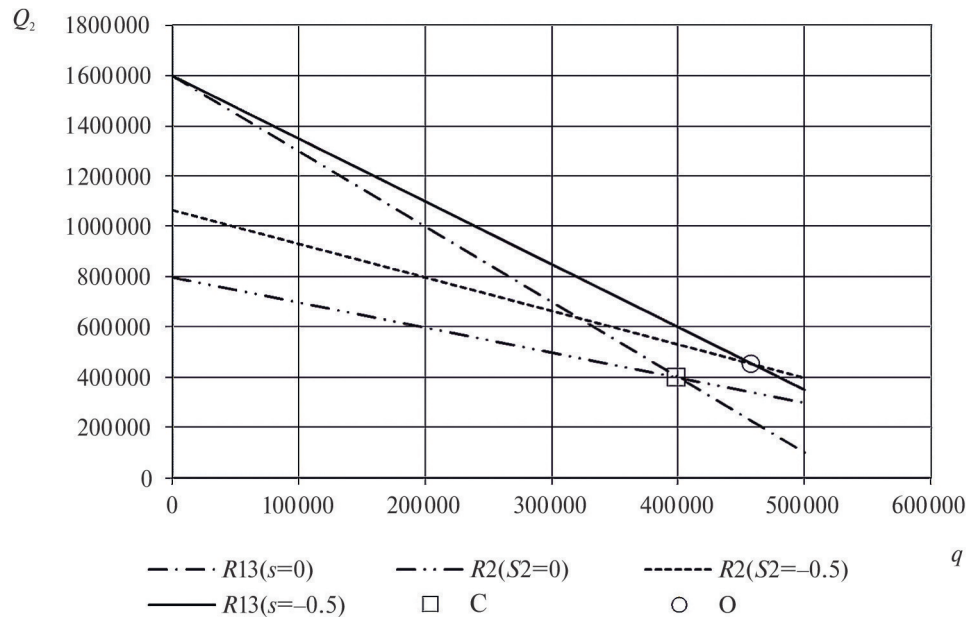


Fig. 1. Cournot equilibrium (C) and three-party Stackelberg leadership (O).

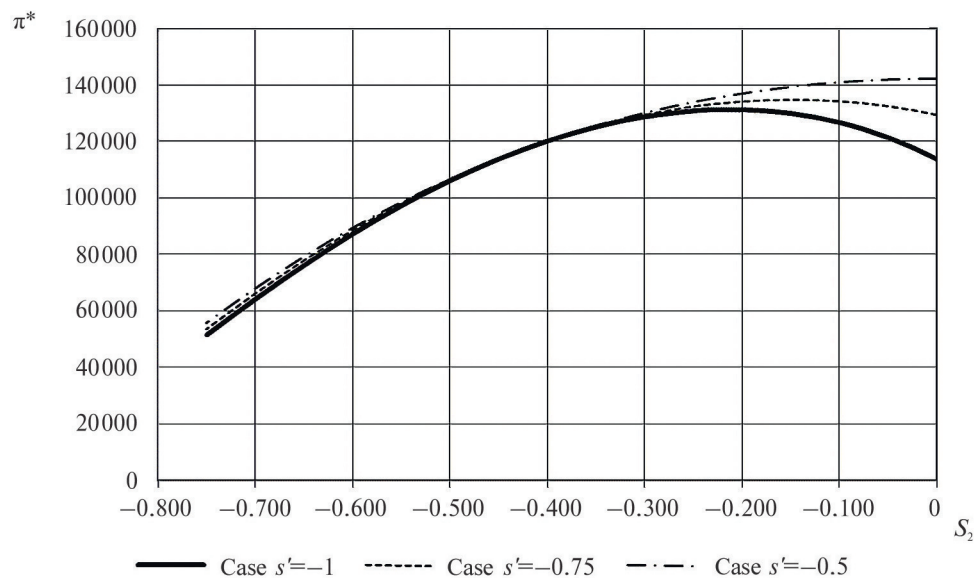


Fig. 2. The environment's utility functions depending on the controlled player's SCV value for different s' .

Table 1 and Fig. 2 show the impact of changes in the environment's response on the corresponding changes in the controlled player's response, expressed by s' . The following model dependence was used:

$$s = -1 + s'S_2,$$

in which $s \in (-1, 0]$ for $S_2 \in (-1, 0]$ and $s' \in (-1, 0]$. According to the data in Table 1, reducing the rate of change of the environment's SCV value (the absolute value of s') shifts the target SCV value $\bar{S}_2$ of the controlled player closer to zero and increases the maximum utility π^* of the environment. The graphs in Fig. 2 demonstrate the existence of a maximum of the environment's utility function for certain SCV values of the controlled player. Note that the second-order optimality condition (8d) was also verified.

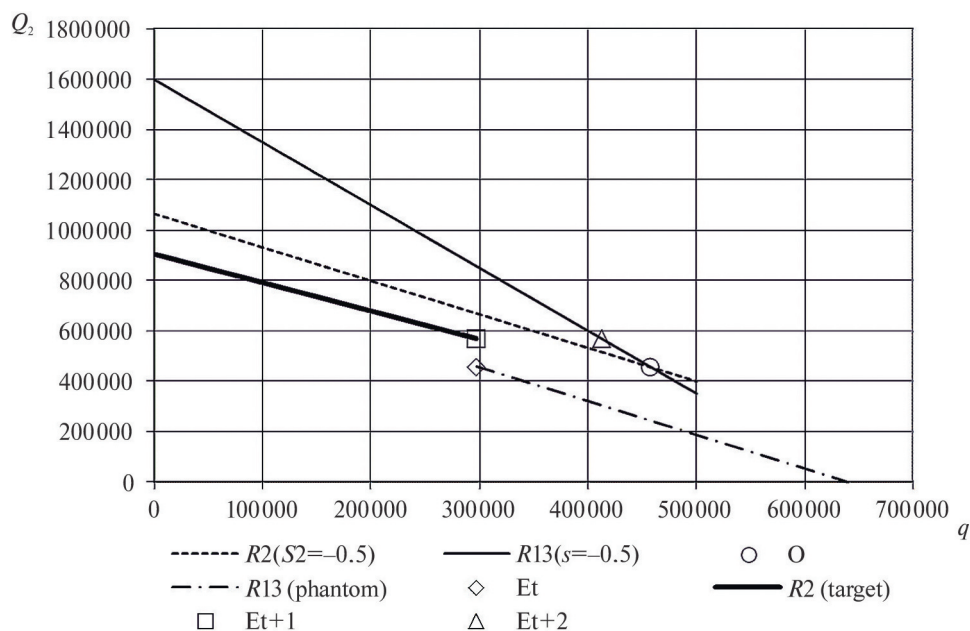


Fig. 3. The informational control process from Stackelberg leadership (0).

Figure 3 demonstrates the informational control process from the state of three-party Stackelberg leadership (point 0) for the case $s' = -1$. At step 1, the environment's action according to its phantom reaction function $R13(\text{phantom})$ transfers the game to the state E^t ; and the controlled player performs an action corresponding to the initial position 0 in the game. This change in the environment's action allows the controlled player to establish that the former's behavior corresponds to the SCV value $\bar{S}_2$; therefore, at step 2, this player changes his/her reaction function to the one desired for the environment, $R2(\text{target})$. Then the game evolves to the state $E^{(t+1)}$, and the environment's action remains at the level E^t . Responding at step 3, the environment returns to its true reaction function $R13(s = -0.5)$, since the environment calculated its optimal target SCV value of the controlled player precisely for its initial SCV value. As a result, the game passes into the state $E^{(t+2)}$. The further process of restoring the initial equilibrium at point 0 is not shown here: at step $t + 2$, the players' actions are already close to this point.

The economic effect of informational control is presented in Table 2 at a discount rate of 10%. The total discounted utility of the environment for three steps of the control process exceeds the total discounted value of its utility in the initial state of the game, which confirms the validity of

Table 1. The impact of changes in the environment's SCV value on the environment's optimum when changing the controlled player's SCV value

Case	s'	$\bar{S}_2$	q^*	π^*
1	-1	-0.2338	5 079 027	128 969
2	-0.75	-0.1478	518 714	134 506
3	-0.5	-0.002	533 454	142 260

Table 2. The economic effect of informational control, $\rho = 0.1$

Step	$e^{-\rho t}$	Nominal values		Discounted values	
		π^{*0}	π^{*t}	π^{*0}	π^{*t}
t	0.905	104 761	163 474	94 821	147 963
$t + 1$	0.819	104 761	129 758	85 823	106 301
$t + 2$	0.741	104 761	85 005	77 680	63 031
Sum				258 324	317 295

such control from the environment's standpoint. Note an increased utility of the environment at step t , which is not always the case; in the situation under consideration, the first action of the environment was less than in the initial equilibrium, resulting in an equilibrium price increase; if the first action of the environment were the opposite, its utility would decrease.

6. CONCLUSIONS

This paper has demonstrated the fundamental possibility of implementing informational control of player's actions in a triopoly game by other players (the environment). The linear models of demand and cost functions have been considered; in this case, explicit expressions for the optimal reactions and equilibrium actions of players have been derived, an informational control algorithm has been developed, and the effect of informational control has been illustrated. The following key results have been obtained.

The dependence of the maximum of the player's utility function on his/her equilibrium action under the reflexive behavior of all players has been established, showing that the player's payoff depends both on his/her SCV value and the SCV value of the environment through the former's equilibrium action. Hence, by varying the SCV value of the environment, this player can increase his/her maximum payoff, which is the basis to control the environment through its SCV value as the control parameter. However, one player cannot manipulate the other players, while a group of players, acting in concert, can change the beliefs of one player.

Therefore, a hierarchical game has been investigated in which the environment controls a certain player by finding the optimal SCV value for him/her. An equation has been obtained for calculating the optimal SCV value of this player in terms of the environment's utility function. To motivate the controlled player to choose the required SCV value, the method of optimal reactions has been used, with explicit formulas derived in the triopoly case. In the three-player game (one controlled player and two opponents with a common goal), equilibrium actions have been determined, and an equation has been compiled for calculating the optimal SCV value of the controlled player.

In the triopoly model under consideration, a phantom reaction-based method has been developed to determine the environment's action inducing the controlled player to use the optimal SCV value from the environment's standpoint. This method has been developed into an algorithm to model the action sequence of the environment and the controlled player in the informational control process; the effectiveness of this process has been assessed.

Computer simulations have been carried out to demonstrate the effectiveness and economic impact of the control algorithm. Although the controlled player receives higher profit only at two steps of the game, this outcome is quite significant from a practical point of view. In real business, the game moment often corresponds to the publication period of the company's financial statements (annually). Therefore, one or two years in which environment players receive predominant payoffs can lead to large financial losses for the controlled player and even force the latter out from the market. Consequently, the results of this study have practical value.

REFERENCES

1. Cournot, A.A., *Researches into the Mathematical Principles of the Theory of Wealth*, London: Hafner, 1960. (Original 1838.)
2. Stackelberg, H., *Market Structure and Equilibrium*, 1st ed. Translation into English, Bazin, Urch & Hill, Springer, 2011. (Original 1934.)
3. Algazin, G.I. and Algazina, D.G., Reflexion Processes and Equilibrium in an Oligopoly Model with a Leader, *Autom. Remote Control*, 2020, vol. 81(7), pp. 1258–1270.

4. Algazin, G.I. and Algazina, Y.G., To the Analytical Investigation of the Convergence Conditions of the Processes of Reflexive Collective Behavior in Oligopoly Models, *Autom. Remote Control*, 2022, vol. 83, no. 3, pp. 367–388.
5. Algazin, G.I. and Algazina, D.G., Modeling the Dynamics of Collective Behavior in a Reflexive Game with an Arbitrary Number of Leaders, *Informatics and Automation*, 2022, vol. 21, no. 2, pp. 339–375.
6. Algazin, G.I. and Algazina, D.G., Aggregative Oligopoly Games with Linear Demand and Cost Functions of Agents, in *MAK: Matematiki – Altaiskomu krayu* (MAK: Mathematicians to Altaiskii Krai), Barnaul, 2024, no. 6, pp. 99–101.
7. Xiao, Y., Zhang, S., and Peng, Y., Dynamic Investigations in a Stackelberg Model with Differentiated Products and Bounded Rationality, *J. Comput. Appl. Math.*, 2022, vol. 414, art. no. 114409.
8. Ougolnitsky, G. and Gorbaneva, O., Sustainability of Intertwined Supply Networks: A Game-Theoretic Approach, *Games*, 2022, vol. 13, no. 3, art. no. 35.
9. Ougolnitsky, G.A. and Usov, A.B., The Interaction of Economic Agents in Cournot Duopoly Models under Ecological Conditions: A Comparison of Organizational Modes, *Autom. Remote Control*, 2023, vol. 84, no. 2, pp. 175–189.
10. Al-Khedhairi, A., Dynamical Study of Competition Cournot-like Duopoly Games Incorporating Fractional Order Derivatives and Seasonal Influences, *Int. J. Nonlinear Sci. Numer. Simulation*, 2020, vol. 21, no. 3–4, pp. 339–359.
11. Zouhar, J. and Zouharova, M., Stackelberg versus Cournot Duopoly with Asymmetric Costs: Primary Markups, Entry Deterrence, and a Comparison of Social Welfare and Industry Profits, *Econom. Theory Bulletin.*, 2020, vol. 8, pp. 89–96.
12. Collie, D.R., Taxation under Oligopoly in a General Equilibrium Setting, *J. Public Economic Theory*, 2019, vol. 21, no. 4, pp. 738–753.
13. Ferrarese, W., When Multiple Merged Entities Lead in Stackelberg Oligopolies, *Rev. Industr. Organization*, 2020, vol. 56, no. 1, pp. 131–142.
14. Mukherjee, A. and Zeng, C., Social Desirability of Entry in a Bilateral Oligopoly – The Implications of (Non) Sunk Costs, *Mathem. Social Sci.*, 2022, vol. 118, pp. 12–19.
15. Zhou, J., Mixed Bundling in Oligopoly Markets, *J. Econom. Theory*, 2021, vol. 194, art. no. 105257.
16. Shuai, J., Yang, H., and Zhang, L., Dominant Firm and Competitive Bundling in Oligopoly Markets, *Games Econom. Behav.*, 2022, vol. 132, pp. 421–447.
17. Zhang, Y., When Should Firms Choose a Risky New Technology? An Oligopolistic Analysis, *Econom. Modelling*, 2020, vol. 91, pp. 687–693.
18. Novikov, D.A. and Chkhartishvili, A.G., *Reflexion and Control: Mathematical Models*, Leiden: CRC Press, 2014.
19. Geraskin, M.I., The Properties of Conjectural Variations in the Nonlinear Stackelberg Oligopoly Model, *Autom. Remote Control*, 2020, vol. 81, no. 6, pp. 1051–1072.
20. Geraskin, M.I., Informational Control of Strategies in an n -Player Oligopoly Game with Reflexive Behavior, *Autom. Remote Control*, 2025, vol. 86, no. 6, pp. 577–588.
21. Geraskin, M.I., Quantity Conjectural Variations in Oligopoly Games under Different Demand and Cost Functions and Multilevel Leadership, *Autom. Remote Control*, 2024, vol. 85, no. 7, pp. 711–724.
22. Geraskin, M.I. and Moiseeva, K.S., Analysis of Demand and Cost Trends of Telecommunications Market Operators in Russia, *Applied Mathematics and Control Sciences*, 2023, no. 3, pp. 157–174.

This paper was recommended for publication by A.G. Chkhartishvili, a member of the Editorial Board

A Modification of the Multidimensional Fuzzy Controller with the Local Term Division of Input Variables

A. F. Antipin^{*,a}, E. V. Antipina^{**,b}, and S. A. Mustafina^{**,c}

^{*}Ufa University of Science and Technology, Sterlitamak branch, Sterlitamak, Russia

^{**}Ufa University of Science and Technology, Ufa, Russia

e-mail: ^aandrejantipin@ya.ru, ^bstepashinaev@ya.ru, ^cmustafina-sa@mail.ru

Received March 25, 2024

Revised February 16, 2025

Accepted February 26, 2025

Abstract—This paper considers a novel modification of the multidimensional fuzzy controller with interrelated adjustable parameters. The block diagram of the controller and expressions for calculating its main parameters are provided. A transition mechanism from the general term division of input variables to their local one with respect to output variables is described. The authors' special software for the visual design of the fuzzy controller, flexible term adjustment, and the assessment and analysis of the system of production rules is presented. Theoretical results are illustrated by examples.

Keywords: multidimensional fuzzy controller, system of production rules, modification, software, term division

DOI: 10.31857/S0005117925070057

1. INTRODUCTION

Currently, fuzzy controllers and their modifications, including hybrid ones, are increasingly applied in the development of automated control systems (ACSs). They have more flexible adjustment compared to traditional PID controllers, which is especially important for multidimensional and multivariable systems with a complex mathematical model or without any mathematical model as such. Fuzzy logic is used to design expert and intelligent control systems [1–7].

The practical implementation issues of fuzzy controllers were considered by several authors. In this context, we note the following drawbacks [8–14]:

- 1) the high complexity of fuzzy control algorithms (including fuzzification and defuzzification);
- 2) the ambiguous choice of an appropriate membership function of continuous variables;
- 3) significant (in some cases, multiple) growth of the number of production rules with increasing the number of input variables;
- 4) no flexibility in the implementation of unconventional approaches to fuzzy controllers, etc.

In addition, third-party software for programmable logic controllers (PLCs), particularly MATLAB [15–17], is often used in the design and program implementation of fuzzy controllers. This requires additional time and funding as well as complicates the introduction of changes during ACS operation.

To settle these issues, it is necessary to improve the characteristics of fuzzy controllers and their operation algorithms; propose new modifications associated with the formation of a system of production rules; and develop special software with the generation of program blocks of fuzzy controllers for modern programming environments for PLCs.

The aim of this paper is to develop, first, a novel modification of the multidimensional fuzzy controller in order to eliminate the above drawbacks and issues and, second, software for adjusting its parameters and the system of production rules.

2. DESCRIPTION OF THE FUZZY CONTROLLER MODIFICATION

We propose the following modification of the fuzzy controller, intended for ACSs with a large number of production rules describing the operation logic of a complex technological object without an adequate mathematical model. The modification is to settle the issues associated with the compilation of a system of production rules (SPRs), eliminate the redundancy of rules, and reduce their maximum number. This is achieved by determining the interrelations of variables and the transition from the general to local term division of input variables.

Figure 1 shows the block diagram of this modification of the multidimensional fuzzy controller with interrelated controlled parameters, with the following notation: $F_1, \dots, F_n$ are the fuzzifiers of input variables x ; $D_1, \dots, D_m$ are the defuzzifiers of output variables y ; FB and DB are fuzzification and defuzzification blocks, respectively, that control the operation of fuzzifiers and defuzzifiers; LIB is a logical inference block of the controller with a production rule block PRB; T_t are the terms of variables handled by LIB at time t ; PrB is a block for predicting the terms of input variables x ; FEB are blocks for expanding the controller's functions, including communication and data exchange with external devices.

Let the term set T^x of a continuous variable x have the form

$$T^x = \left\{ T_1^x \left(x_1^{(l)} \leq x_t < x_1^{(r)} \right), T_2^x \left(x_2^{(l)} \leq x_t < x_2^{(r)} \right), \dots, T_{k(x)}^x \left(x_{k(x)}^{(l)} \leq x_t \leq x_{k(x)}^{(r)} \right) \right\},$$

where $x^{(l)}$ and $x^{(r)}$ are values determining the left and right limits of the working ranges of the terms T ; x_t is the value of the variable x at a time t ; $k(x)$ is the total number of terms.

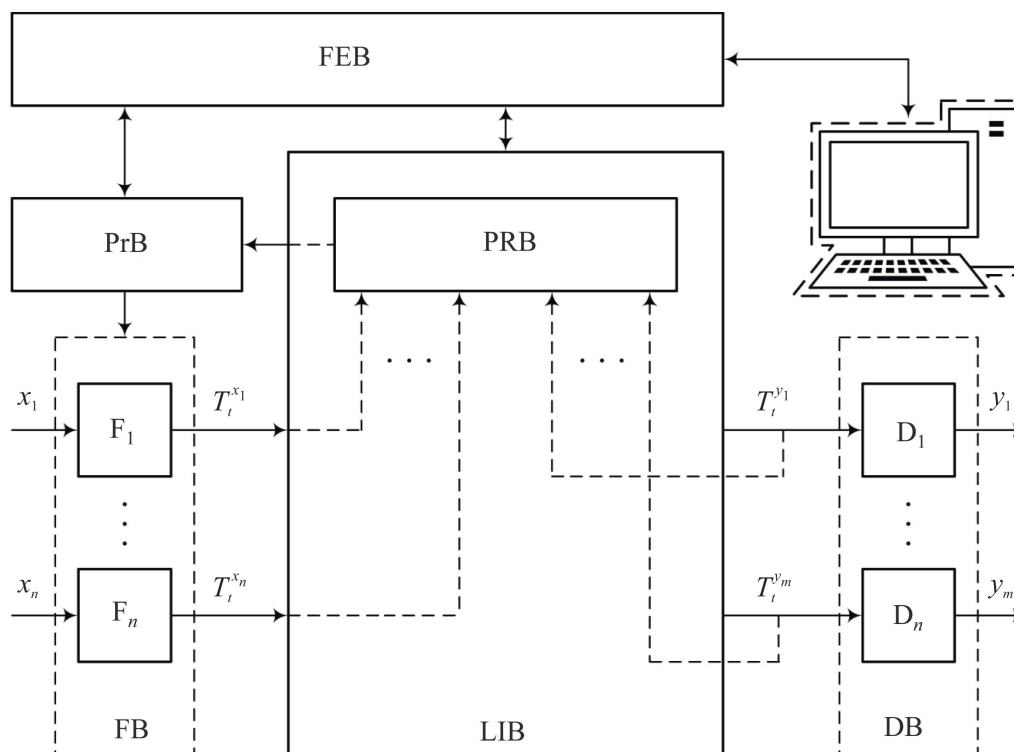


Fig. 1. The block diagram of the fuzzy controller modification.

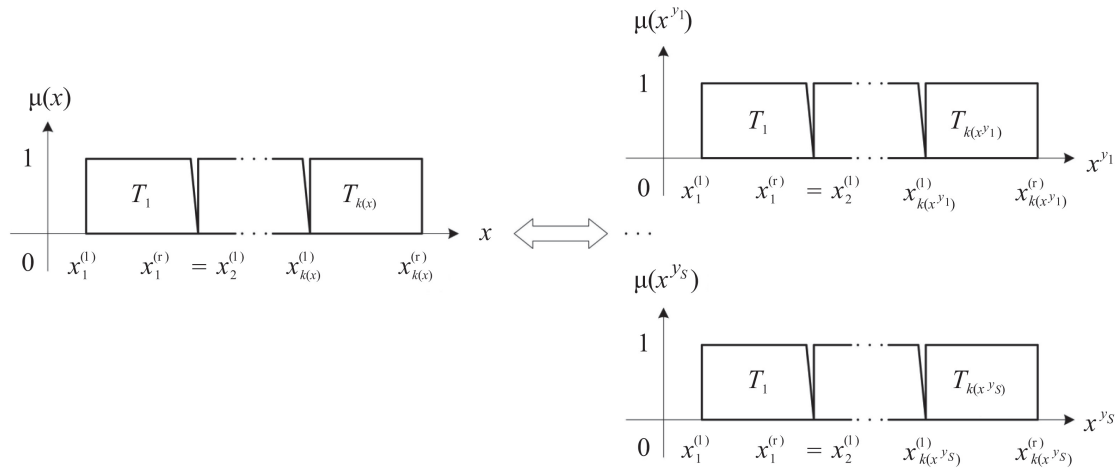


Fig. 2. The division of the general term set into local term sets.

To improve illustrativeness, we choose the class of trapezoidal (close to rectangular) membership functions of x , which is characterized by the empty intersection of the ranges of terms within the general term division of the variable.

By definition,

$$x_k^{(l)} = x_{k-1}^{(r)},$$

and therefore, the value of one range limit is sufficient to describe $k(x) - 1$ terms T . The limits of the first or last term are described directly by two values. For example,

$$T^x = \left\{ T_1^x \left(x_1^{(l)} \right), T_2^x \left(x_2^{(l)} \right), \dots, T_{k(x)}^x \left(x_{k(x)}^{(l)} \leq x_t \leq x_{k(x)}^{(r)} \right) \right\}.$$

As is known, the maximum number $R_{\max}$ of the controller's production rules depends on the total number of the terms of its input variables [18], i.e.,

$$R_{\max} = \prod_{i=1}^n k(x_i),$$

where n specifies the number of the controller's input variables.

In the presence of output variables with feedback (see Fig. 1), the expression for calculating $R_{\max}$ takes the form

$$R_{\max} = \prod_{i=1}^n k(x_i) \prod_{j=1}^o k(y_j),$$

where o is the number of output variables of the feedback controller.

For a multidimensional fuzzy controller with m output variables y , the general term division of some input variables x may be redundant for calculating the values of separate output variables and cause defuzzification errors associated primarily with the absence of production rules defined by technical specifications.

The general term division can be simplified to a local one with respect to particular variables y . For instance, the local term set T^{xy} of a continuous variable x with respect to an interrelated variable y can be analytically represented as follows:

$$T^{xy} = \left\{ T_1^{xy} \left(x_1^{y(l)} \leq x_t < x_1^{y(r)} \right), \dots, T_{k(xy)}^{xy} \left(x_{k(xy)}^{y(l)} \leq x_t \leq x_{k(xy)}^{y(r)} \right) \right\},$$

where $k(x^y) \leq k(x)$.

This process is visualized in Fig. 2.

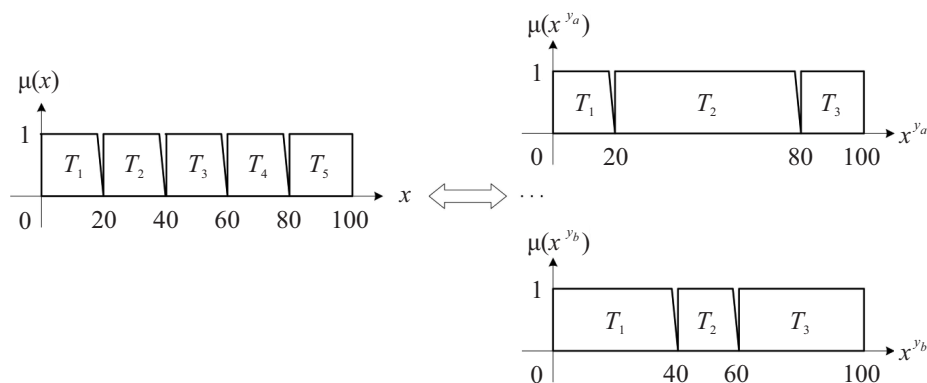


Fig. 3. The conflict of local terms when interpreting variables.

In this case, the expression for calculating the maximum number $R'_{\max}$ of the controller's production rules takes the form

$$R'_{\max} = \sum_{j=1}^m R_{\max}^{y_j} = \sum_{j=1}^m \prod_{i=1}^s k(x_i^{y_j}),$$

where $R_{\max}^y$ is the maximum number of production rules for a particular output variable y ; s denotes the number of input variables x interrelated with the output variable y .

To perform calculations, for the sake of convenience, we present the initial data on the number of the terms of input variables x for their interrelated output variables y in Table 1. Here, the symbol “—” in particular cells indicates the absence of interrelation between the corresponding variables.

Table 1. Data to analyze the term relations of variables

Output variable	The number of the terms of input variables		
	x_1	...	x_n
y_1	$k(x_1^{y_1})$	...	$k(x_n^{y_1})$
...	...	...	...
y_m	$k(x_1^{y_m})$	...	$k(x_n^{y_m})$

In this case, the minimum number of the terms of output variables must satisfy the condition

$$k_{\min}(y) \leq R^y,$$

where R^y is the number of production rules for a particular output variable y .

If the number R^y of production rules is less than the total number of the terms of y , some terms will remain unused (not participate in forming the system of production rules). Such terms should be removed by uniting with neighbor ones to avoid gaps within the working ranges of the output variables. If the union of terms is impossible, such variables cannot be used for feedback implementation in order to avoid fuzzification errors.

This approach allows eliminating the so-called conflict of local terms due to the overlap of their ranges when forming the general term division (Fig. 3), which leads to the emergence of new terms and consequently increases the maximum number of production rules of the controller.

According to Fig. 3, the general term division of a continuous variable x ,

$$T^x = \{T_1^x(0), T_2^x(20), T_3^x(40), T_4^x(60), T_5^x(80 \leq x_t \leq 100)\},$$

is formed using the following local term divisions of x with respect to the output variables y_a and y_b :

$$\begin{aligned} T^{x^{y_a}} &= \{T_1^{x^{y_a}}(0), T_2^{x^{y_a}}(20), T_3^{x^{y_a}}(80 \leq x_t \leq 100)\}, \\ T^{x^{y_b}} &= \{T_1^{x^{y_b}}(0), T_2^{x^{y_b}}(40), T_3^{x^{y_b}}(60 \leq x_t \leq 100)\}. \end{aligned}$$

For instance, the term T_1^x corresponds to the term $T_1^{x^{y_a}}$; the term T_3^x , to the term $T_2^{x^{y_b}}$; the term T_5^x , to the term $T_3^{x^{y_a}}$. The terms T_2^x and T_4^x are obtained by uniting the ranges of the remaining terms from the local term divisions of the continuous variable x with respect to the output variables y_a and y_b .

In this example, the general term division of the continuous variable x leads to the following problems. The terms T_1^x and T_2^x , T_4^x and T_5^x duplicate the production rules for the output variable y_b , and the terms T_2^x , T_3^x , and T_4^x do the same for the output variable y_a . This affects the total number of production rules, especially when developing multivariable fuzzy systems with

$$y = f(x_1, \dots, x_s).$$

Moreover, uniting terms can lead to the incorrect operation of defuzzifiers with certain classes of membership functions (e.g., triangular) and several difficulties when forming a system of production rules.

Local term division resolves these problems, allowing the engineer to develop a fuzzy ACS in full compliance with technical specifications.

If necessary, critical production rules can be identified: they involve the terms of continuous variables marked as “critical” to indicate their particular importance (e.g., when determining the extreme ranges of values as an emergency signal).

The number of critical rules for each group of interrelated variables, $R_{\text{gr cr}}^g$, can be calculated as follows:

$$R_{\text{gr cr}}^g = R_{\text{gr}}^g - \prod_{i=1}^{n_g} (k(x_i) - k_{\text{cr}}(x_i)),$$

where k_{cr} is the number of critical terms defined for input variables x in the volume n_g , forming a group with the ordinal number g ; R_{gr}^g is the maximum number of production rules of the group:

$$R_{\text{gr}}^g = \prod_{i=1}^{n_g} k(x_i).$$

3. DESCRIPTION OF THE CONTROLLER SOFTWARE

Modern design and development environments for fuzzy controllers have no capabilities for determining the interrelations of variables and passing from the general term division of input variables to local ones with respect to output variables, as well as no deep analysis tools for a system of production rules. Therefore, we developed a software solution (computer-aided system) for the visual development of a PLC program for the above modification of the multidimensional fuzzy controller, the calculation and analysis of its main parameters, the formation of production rules, etc.; see Figs. 4 and 5.

When creating a new fuzzy controller project, the following basic information is entered: the number of input and output variables, their working ranges, and interrelations. Terms are modified, and production rules are entered, in a special visual editor.

After data entry is complete, the user can analyze the term divisions of variables and the system of production rules. In this case, the total number of terms and the number of critical terms (for the entire controller and each variable separately), the belonging of variables to a particular group (control loop), etc., are assessed.

Fig. 4. The term editor for continuous variables and their interrelations.

Fig. 5. Analysis results for variables and the system of production rules.

```

VAR_INPUT
  NUM: INT;
  ...
  OUT_RM: ARRAY [0..R-1] OF BYTE;
END_VAR

VAR_OUTPUT
  ...
  OUT_M: REAL;
END_VAR

VAR
  ...
  OUT_MM: ARRAY [0..K(OUT_M)] OF REAL;
END_VAR
...

//DEFUZZIFIER M
IF BYTE_TO_INT(OUT_RM[NUM]) > 0
  THEN OUT_M := (OUT_MM[BYTE_TO_INT(OUT_RM[NUM]) -1]
    + OUT_MM[BYTE_TO_INT(OUT_RM[NUM])]) / 2;
END_IF;

```

Fig. 6. SCL program code of the defuzzification functional block.

A separate table displays the values of the controller parameters depending on the content of the system of production rules (Fig. 5):

- 1) the maximum $R_{\max}$ and factual $R'_{\max}$ numbers of production rules forming the system of production rules of the controller;
- 2) the number of groups of interrelated variables;
- 3) the number R_{gr}^g of production rules for each group, the number $R_{\text{gr cr}}^g$ of critical rules, etc.

The settings of the interrelations and term divisions of variables can be placed in a separate file of a special format and loaded into the computer-aided design system developed by the authors [19]. This system contains an editor for the system of production rules and special tools for checking its completeness as well as the correctness of the user data.

Upon completion of the work, it is possible to generate a ready-made program of the controller and/or its separate functional blocks (fuzzifiers and defuzzifiers) in the SCL and AWL format for Simatic S7 PLCs considering the requirements of the IEC 61131-7:2000 standard (Programmable controllers – Part 7: Fuzzy control programming).

Figure 6 presents a fragment of the generated SCL program code of the defuzzification functional block for a continuous output variable M whose crisp value is found by calculating the midpoint of the range of the term active at the current time [18].

4. COMPUTATIONAL EXPERIMENT

We carry out a computational experiment to assess the system of production rules of the fuzzy controller when passing from the general to local term division of input variables.

Consider a control loop with four input and two output continuous variables x and y , respectively; their interrelation diagram and general term divisions in the working range $[0, 100]$ (in simplified form) are shown in Fig. 7. Since the terms of the output quantities do not affect the total number of production rules, they can be omitted for calculations.

The term sets of the input continuous variables have the form

$$\begin{aligned}
 T^{x_1} &= \{T_1(0), T_2(10), T_3(25), T_4(50), T_5(75), T_6(90; 100)\}, \\
 T^{x_2} &= \{T_1(0), T_2(25), T_3(75; 100)\}, \\
 T^{x_3} &= \{T_1(0), T_2(10), T_3(20), T_4(50), T_5(80), T_6(90; 100)\}, \\
 T^{x_4} &= \{T_1(0), T_2(15), T_3(85; 100)\}.
 \end{aligned}$$

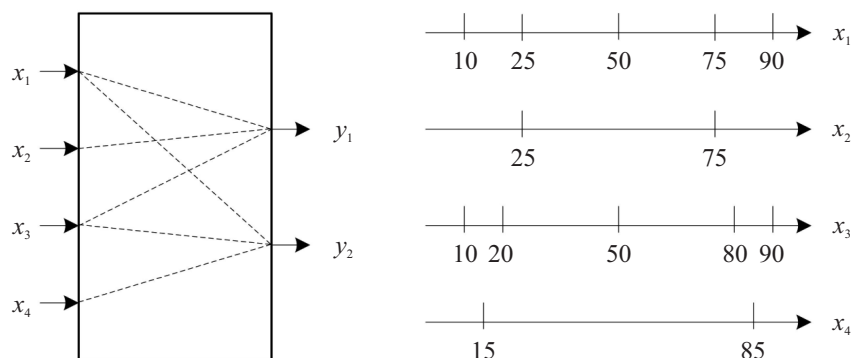


Fig. 7. The interrelation diagram and general term division of input variables.

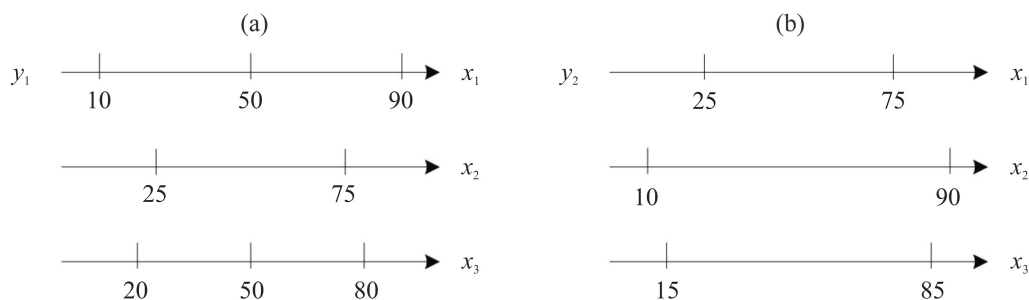


Fig. 8. The local term division of the input variables x_1 , x_2 , x_3 and x_4 with respect to the output variables (a) y_1 and (b) y_2 .

Thus, $k(x_1) = 6$, $k(x_2) = 3$, $k(x_3) = 6$, and $k(x_4) = 3$. The maximum number $R_{\max}$ of production rules is

$$R_{\max} = \prod_{i=1}^4 k(x_i) = 324.$$

Being extremely redundant, this number will strongly complicate the formation of the system of production rules of the controller. However, certain interrelations can be used at the controller design stage to pass from the general to local term division, and production rules can be subsequently introduced for each output variable separately.

As is known, the input variables x_1 , x_2 , and x_3 are related to the output variable y_1 whereas x_1 , x_3 , and x_4 to y_2 . Figure 8 demonstrates their local term division.

The term sets of the continuous input variables x_1 , x_2 , and x_3 with respect to the continuous output variable y_1 have the form

$$\begin{aligned} T^{x_1^{y_1}} &= \{T_1(0), T_2(10), T_3(50), T_4(90; 100)\}, \\ T^{x_2^{y_1}} &= \{T_1(0), T_2(25), T_3(75; 100)\}, \\ T^{x_3^{y_1}} &= \{T_1(0), T_2(20), T_3(50), T_4(80; 100)\}. \end{aligned}$$

Next, the term sets of the continuous input variables x_1 , x_3 , and x_4 with respect to the continuous output variable y_2 have the form

$$\begin{aligned} T^{x_1^{y_2}} &= \{T_1(0), T_2(25), T_3(75; 100)\}, \\ T^{x_3^{y_2}} &= \{T_1(0), T_2(10), T_3(90; 100)\}, \\ T^{x_4^{y_2}} &= \{T_1(0), T_2(15), T_3(85; 100)\}. \end{aligned}$$

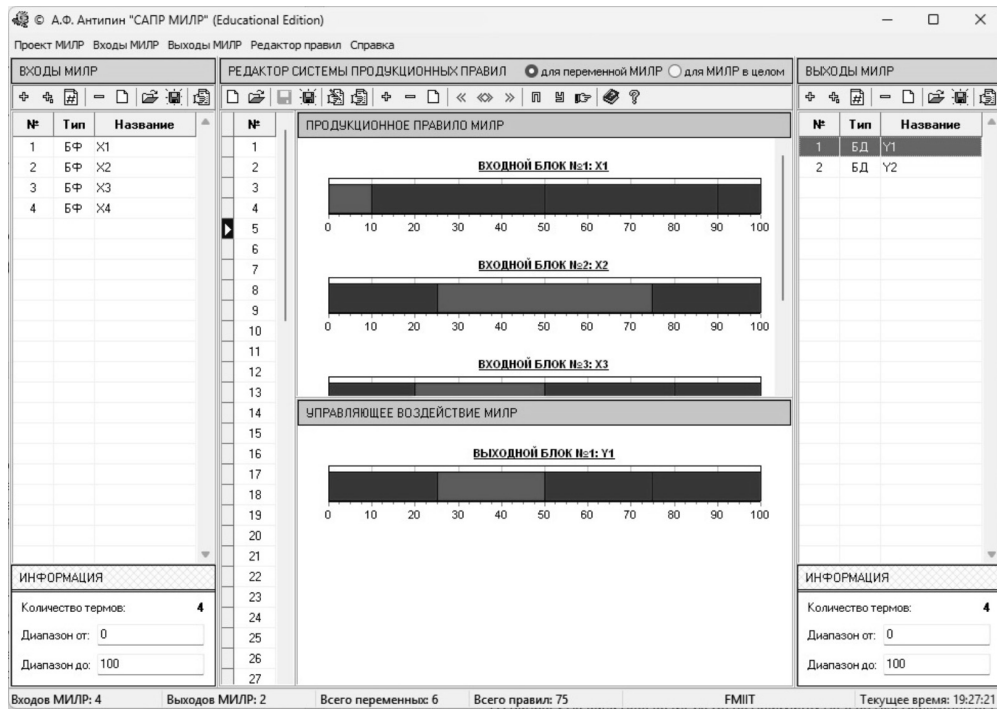


Fig. 9. The editor of the computer-aided design system.

The numbers of the terms of the input variables x for particular output variables y are combined in Table 2.

Table 2. Data to analyze the term relations of variables

Output variable	The number of terms of input variables			
	x_1	x_2	x_3	x_4
y_1	4	3	4	—
y_2	3	—	3	3

Thus, in view of the known interrelations between the input and output variables, the maximum number $R'_{\max}$ of production rules is

$$R'_{\max} = \sum_{j=1}^2 R_{\max}^{y_j} = k(x_1)k(x_2)k(x_3) + k(x_1)k(x_3)k(x_4) = 75.$$

The resulting reduction in the maximum number of production rules, $\Delta R_{\max}$, reaches

$$\Delta R_{\max} = \frac{R_{\max} - R'_{\max}}{R_{\max}} = 0.8$$

(4.3 times).

This process can be automated using the above software solution. Based on the calculation results, it is possible to assess the design complexity and save the settings of the interrelations and term division of the variables for subsequent SPR formation in the computer-aided design system (Fig. 9).

Working in a single visual programming environment simplifies coding for the multidimensional fuzzy controller with interrelated adjustable parameters as well as calculating its basic characteristics.

5. CONCLUSIONS

The novel modification of the multidimensional multivariable fuzzy controller described in this paper provides several advantages in automated control system design: the flexible adjustment of the terms of variables, reduction in the maximum number of production rules, a separate defuzzification function specified for each output variable [20], etc.

The original special software developed by the authors contains tools for the visual design of the fuzzy controller, the evaluation and analysis of the system of production rules, and the generation of a ready program for the Simatic S7 PLC. This software solution can be used by experts and engineers to design automated control systems for complex technological objects without adequate mathematical models and create reliable and effective PLC programs for fuzzy controllers, reducing the time of their development.

FUNDING

This work was carried out within a state order of the Ministry of Science and Higher Education of the Russian Federation (scientific theme code FZWU-2023-0002).

REFERENCES

1. Dumitrescu, C., Ciotirnae, P., and Vizitiu, C., Fuzzy Logic for Intelligent Control System Using Soft Computing Applications, *Sensors*, 2021, vol. 21, no. 8, art. no. 2617. <https://doi.org/10.3390/s21082617>
2. Vasil'ev, V.I. and Il'yasov, B.G., *Intellectual'nye sistemy upravleniya. Teoriya i praktika* (Intelligent Control Systems. Theory and Practice), Moscow: Radiotekhnika, 2009.
3. Kocian, J., Koziorek, J., and Pokorný, M., Implementation of Fuzzy Logic Control Based on PLC, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011, pp. 1–8. <https://doi.org/10.1109/ETFA.2011.6059049>
4. Nguyen, L. and Novák, V., Forecasting Seasonal Time Series Based on Fuzzy Techniques, *Fuzzy Sets and Systems*, 2019, vol. 361, pp. 114–129. <https://doi.org/10.1016/j.fss.2018.09.010>
5. Piegat, A., *Fuzzy Modeling and Control*, Heidelberg: Physica-Verlag, 2001.
6. Boukezzoula, R., Coquin, D., Nguyen, T.L., et al., Multi-sensor Information Fusion: Combination of Fuzzy Systems and Evidence Theory Approaches in Color Recognition for the NAO Humanoid Robot, *Robot. Autonom. Syst.*, 2018, vol. 100, pp. 302–316. <https://doi.org/10.1016/j.robot.2017.12.002>
7. Melin, P., Castillo, O., Kacprzyk, J., et al., *Fuzzy Logic in Intelligent System Design*, Springer, 2018.
8. Filimonov, A.B. and Filimonov, N.B., Certain Problematic Aspects of Fuzzy PID Regulation, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2018, vol. 19, no. 12, pp. 762–769. <https://doi.org/10.17587/mau/19.762-769>
9. Bykovtsev, Y.A. and Lokhin, V.M., Estimation of the Accuracy of a Control System with a Fuzzy PID Controller Based on the Approximation of the Static Characteristic of the Controller, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2021, vol. 22, no. 12, pp. 619–624. <https://doi.org/10.17587/mau.22.619-624>
10. Bobyr, M.V. and Emelyanov, S.G., A Nonlinear Method of Learning Neuro-Fuzzy Models for Dynamic Control Systems, *Appl. Soft. Comput.*, 2020, vol. 8, art. no. 106030. <https://doi.org/10.1016/j.asoc.2019.106030>
11. Karakuzu, C., Karakaya, F., and Pavuslu, M.A., FPGA Implementation of Neuro-Fuzzy System with Improved PSO Learning, *Neural Networks*, 2016, vol. 76, pp. 128–140. <https://doi.org/10.1016/j.neunet.2016.02.004>

12. Zhirabok, A.N., Suvorov, A.Yu., and Shumskii, A.E., Robust Diagnosis of Discrete Systems with Delay: Logic-Dynamical Approach, *J. Comput. Syst. Sci. Int.*, 2014, vol. 53, no. 1, pp. 47–62. <https://doi.org/10.1134/S1064230714010134>
13. Verma, O.P., Jain, V., and Gumber, R., Simple Fuzzy Rule Based Edge Detection, *J. Inform. Proc. Syst.*, 2013, vol. 9, no. 4, pp. 575–591. <https://doi.org/10.3745/JIPS.2013.9.4.575>
14. Yongming, L., Yali, L., and Zhanyou, M., Computation Tree Logic Model Checking Based on Possibility Measures, *Fuzzy Sets and Systems*, 2015, vol. 262, pp. 44–59. <https://doi.org/10.48550/arXiv.1401.4658>
15. Ignatyev, V.V. and Solovev, V.V., A Synthesis Method for Fuzzy Controllers Based on Clustering, *Software & Systems*, 2021, vol. 34, no. 4, pp. 597–607. <https://doi.org/10.15827/0236-235X.136.597-607>
16. Bogachev, I.S. and Kolomytsev, V.G., Technique for Adjusting a Fuzzy Regulator with a Nonlinear P-Type Characteristic by Matlab, *Modern High Technologies*, 2020, no. 11-1, pp. 9–14. <https://doi.org/10.17513/snt.38330>
17. Buriboev, A., Kang, H.K., Ko, M.-C., et al., Application of Fuzzy Logic for Problems of Evaluating States of a Computing System, *Appl. Sci.*, 2019, vol. 9, no. 15, art. no. 3021. <https://doi.org/10.3390/app9153021>
18. Antipin, A.F. and Antipina, E.V., Multidimensional Multi-Connected Fuzzy Interval-Logic Controller, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2024, vol. 25, no. 2, pp. 72–78. <https://doi.org/10.17587/mau.25.72-78>
19. Antipin, A.F. and Antipina, E.V., Visual Programming Environment for Multidimensional Fuzzy Interval-Logic Regulators, *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 5, pp. 1041–1049. <https://doi.org/10.17586/2226-1494-2023-23-5-1041-1049>
20. Antipin, A.F. and Antipina, E.V., Fuzzy Modeling of Technological Processes and Objects Using Multidimensional Interval-Logic Controllers, *Optoelectron. Instrument. Proc.*, 2020, vol. 56, pp. 68–75. <https://doi.org/10.3103/S8756699020010094>

This paper was recommended for publication by R.A. Munasypov, a member of the Editorial Board

Randomized Machine Learning Methods for Generating Random Data Ensembles with Given Numerical Characteristics

Yu. S. Popkov^{*,**,a}, A. Yu. Popkov^{*,b}, and Yu. A. Dubnov^{*,c}

**Federal Research Center “Computer Science and Control,”
Russian Academy of Sciences, Moscow, Russia*

***Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
e-mail: ^apopkov@isa.ru, ^bapopkov@isa.ru, ^cyury.dubnov@phystech.edu*

Received March 25, 2024

Revised February 16, 2025

Accepted May 29, 2025

Abstract—This paper considers the problem of generating random data ensembles with given numerical characteristics. A solution method is developed using randomized machine learning procedures based on a sequence of functional entropy-linear programming problems with constraints in the form of normalized moments. The generation problem is reduced to a system of nonlinear equations with integral components. The authors’ asymptotic analytical method is adapted to transform these equations into a system of equations with a polynomial left-hand side. The analytical methods are applied to generate random data ensembles for asset price dynamics.

Keywords: entropy, randomization, normalized moments, random ensembles, machine learning, power series, analytic functions, multidimensional integrals, multidimensional polynomials

DOI: 10.31857/S0005117925070069

1. INTRODUCTION

Machine learning methods for many applied and scientific-practical problems are based on data with required properties. In the context of machine learning methods of the statistical approach, data requirements are implemented in the form of their probabilistic and numerical characteristics [1]. In addition to traditional classification and forecasting [2–7], we mention other areas such as software testing [8–10] and knowledge control [11–13], where data serve either to train process models or evaluate hypotheses statistically. In parallel, the so-called scenario approach is being developed: parameter arrays are compiled for a parameterized process model (most often, by experts), and corresponding data ensembles are generated [14, 15]. In any case, data properties must be properly considered for the correct use of the corresponding theoretical methods and approaches as well as in the practical application of the developed and trained models.

Nowadays, rich data are available in many areas, as they are collected in abundance and accumulated automatically during the operation of information systems and various technical devices. At the same time, a still open challenge is to generate necessary data (with required properties) for the purpose of developing, training, and testing methods and devices. Of course, the required properties can be formalized in different ways in particular areas and problems. In this paper, such a formalization is performed within a probabilistic concept; in particular, by suitable data we mean random ensembles with appropriate probability density functions (PDFs). By assumption,

appropriate PDFs can be sampled, in one way or another, i.e., transformed into corresponding random sequences.

Thus, suitable data ensembles are generated by reconstructing optimal, in an accepted sense, PDFs considering given requirements. Rather high uncertainty arises when formulating such requirements, and entropy is a natural criterion for optimizing PDFs [16–20]. However, this is not enough, and some additional properties of PDFs are often necessary. Some of them can be formulated in terms of numerical characteristics, namely, moments, semi-invariants, etc. Therefore, the construction of desired PDFs is reduced to constrained maximization of an information entropy functional. In its formal description, this problem is close to some mathematical models of randomized machine learning (RML) studied in [21, 22]. Certain differences are associated with the system of constraints in the problem under consideration.

In this paper, we further develop RML methods in the following directions:

- randomized learning under additional moment-type constraints;
- adaptation of the analytical method for solving nonlinear equations with integral components to the problem under consideration;
- effectiveness analysis of the methods in asset price forecasting as an illustrative example.

Note that RML problems and the problem of constructing desired distributions involve essentially nonlinear equations with the so-called integral components. They represent multidimensional integrals with exponential subintegral functions that are defined on simple sets (parallelepipeds) and parameterized by Lagrange multipliers. Using the analytic properties of exponential functions, these integrals are approximated by parameterized integrals of multidimensional polynomials as integrand functions. The latter are calculated analytically.

With the indicated transformations of multidimensional parameterized integrals, a system of nonlinear equations with integral components is approximated by a system of equations with a polynomial left-hand side. They are solved by an analytical method based on abstract power series [23, 24].

In view of the aforesaid, the remainder of this paper is organized as follows. Section 2 presents the general problem statement addressed. The solution approach and necessary theoretical tools are described in Section 3. Next, Section 4 is devoted to asset price forecasting. Section 5 discusses the features of the results and directions for further research. Finally, the outcomes of this paper are summarized in Section 6.

2. PROBLEM STATEMENT

Consider a random sequence $u[n]$, where $n \in \mathcal{N} = \overline{1, N}$, and let $Y_{(s \times N)}$ be a given data matrix of numerical characteristics whose elements describe the values of *normalized moments*¹ of order $k = \overline{1, s}$ at observation points (time instants) $n = \overline{1, N}$:

$$Y_{(s \times N)} = \left\{ \left(\mathcal{M}\{u^k[n]\} \right)^{1/k} \right\} = \left\{ y^k[n] \mid k = \overline{1, s}, n = \overline{1, N} \right\}. \quad (1)$$

In particular, this kind of information often appears in the forecasting of financial instruments prices at stock exchanges [26–28].

The problem of generating data with given properties can be formulated as follows:

At each time instant n , it is required to generate ensembles $\mathcal{Z}_n$ of random sequences $z[n]$, $n = \overline{1, N}$, with s normalized moments

$$m^{(k)}[n] = \left(\mathcal{M}\{z^k[n]\} \right)^{1/k}, \quad k = \overline{1, s}, \quad (2)$$

that equal given normalized moments $y^k[n]$ (1).

¹ Here, normalized moments are selected as numerical characteristics. But it is possible to consider, e.g., semi-invariants or the mathematical expectations of continuous functions of a random sequence.

The generator of the ensemble $\mathcal{Z}_n$ is an *input-output model* or an *auto-model*.² In both cases, the model has random parameters $\mathbf{a} \in \mathcal{A} \subset R^r$ of the interval type:

$$\mathbf{a} \in \mathcal{A} \subset R^r, \quad \mathcal{A} = [\mathbf{a}^-, \mathbf{a}^+]. \quad (3)$$

For each time instant n , they are characterized by a continuously differentiable PDF $P_{(n)}(\mathbf{a})$.

Depending on the availability of a priori information about the data origin, either a static or dynamic input-output model is used.

A *static input-output model* generating random sequences $z[n]$ is characterized by a nonlinear differentiable function φ with parameters $\mathbf{a}$:

$$z[n|\mathbf{a}] = \varphi(\mathbf{x}[n]|\mathbf{a}), \quad n = \overline{1, N}, \quad (4)$$

where $\mathbf{x}[n] = \{x_1[n], \dots, x_m[n]\}$ and z are the input and output of this model.

From this point onwards, the symbol “|” indicates that random parameters $\mathbf{a}$ with a PDF $P_{(n)}(\mathbf{a})$ are realized for each time instant n .

A *dynamic model* is characterized by a continuous nonlinear functional $\mathcal{B}$:

$$z[n|\mathbf{a}] = \mathcal{B}[\mathbf{x}[\tau], n-p \leq \tau \leq n|\mathbf{a}], \quad n = \overline{1, N}, \quad (5)$$

where p means the model memory (the number of previous input values affecting the current output value). The random parameters $\mathbf{a}$ are of the interval type (3).

This problem is solved in two stages. The *first* stage is to find the optimal probabilistic characteristics of the random parameters, namely, the PDFs $P_{(n)}(\mathbf{a})$ for all $n = \overline{1, N}$ in the static model (4) or for all $n = \overline{1-p, N}$ in the dynamic model (5). The *second* stage consists in the sampling of these PDFs, i.e., transforming them into corresponding random ensembles $\mathcal{Z}_n$.

3. MATERIALS AND METHODS

3.1. Optimization of PDFs of Model Parameters

To solve the first-stage problem, we utilize the methodology of randomized machine learning [22]: the specified sequences will be generated by a mathematical input-output model with random parameters optimized by the information entropy criterion.

It is convenient to introduce the following s -dimensional vectors for further considerations:

- the vector of given normalized moments for each n ,

$$\mathbf{y}^{(n)} = \{y_1[n], \dots, y_s[n]\}; \quad (6)$$

- the model output vector for each n ,

$$\mathbf{z}^{(n)}(\mathbf{a}) = \{z_1[n|\mathbf{a}], \dots, z_s[n|\mathbf{a}]\}; \quad (7)$$

- the vector of model output's normalized moments for each n ,

$$\mathbf{m}^{(n)} = \left\{ \int_{\mathcal{A}} P_{(n)}(\mathbf{a}) z[n|\mathbf{a}] d\mathbf{a}, \dots, \left(\int_{\mathcal{A}} P_{(n)}(\mathbf{a}) z^s[n|\mathbf{a}] d\mathbf{a} \right)^{1/s} \right\}. \quad (8)$$

According to [22], the problem of finding the optimal PDFs $P_{(n)}(\mathbf{a})$ can be formulated for each $n = \overline{1, N}$ as follows: maximize the information entropy functional

$$\mathcal{H}_{(n)}[P_{(n)}(\mathbf{a})] = - \int_{\mathcal{A}} P_{(n)}(\mathbf{a}) \ln P_{(n)}(\mathbf{a}) d\mathbf{a} \Rightarrow \max \quad (9)$$

² Here, we use an input-output model. In the illustrative example below, an auto-model is described by difference equations.

subject to the constraints

$$\bullet \quad \int_{\mathcal{A}} P_{(n)}(\mathbf{a}) d\mathbf{a} = 1 \quad (10)$$

(the normalization conditions) and

$$\bullet \quad \mathbf{m}^{(n)} = \mathbf{y}^{(n)} \quad (11)$$

(the balances of the model output's normalized moments with the data).

Assuming that the PDFs are continuously differentiable, the solution parameterized by the Lagrange multipliers $\boldsymbol{\lambda}^{(n)} = \{\lambda_1^{(n)}, \dots, \lambda_s^{(n)}\}$ has the form [22]

$$P_{(n)}^*(\mathbf{a}) = \frac{\exp\left(-\langle \boldsymbol{\lambda}^{(n)}, \mathbf{z}^{(n)}(\mathbf{a}) \rangle\right)}{\mathbb{P}_{(n)}(\boldsymbol{\lambda}^{(n)})}, \quad (12)$$

where

$$\mathbb{P}_{(n)}(\boldsymbol{\lambda}^{(n)}) = \int_{\mathcal{A}} \exp\left(-\langle \boldsymbol{\lambda}^{(n)}, \mathbf{z}^{(n)}(\mathbf{a}) \rangle\right) d\mathbf{a} \quad (13)$$

and $\langle \bullet, \bullet \rangle$ denotes the inner product of vectors with s components.

The Lagrange multipliers $\boldsymbol{\lambda}^{(n)}$ satisfy the following system of s equations, called the *balance* equations:

$$\int_{\mathcal{A}} \exp\left(-\langle \boldsymbol{\lambda}^{(n)}, \mathbf{z}^{(n)}(\mathbf{a}) \rangle\right) [\mathbf{z}^{(n)}(\mathbf{a}) - \mathbf{y}^{(n)}] d\mathbf{a} = \mathbf{0}. \quad (14)$$

This system determines the vectors of the Lagrange multipliers $\boldsymbol{\lambda}^{(n)}$ for each time instant n from the interval $[1, N]$.

The second-stage problem (the transformation of the entropy-optimal PDF into the corresponding random sequence) can be implemented using the methods described in [25].

3.2. An Analytical Method for Calculating Multidimensional Integrals

In view of equation (14), to determine the optimal PDF functions, it is necessary to calculate multidimensional integrals and then solve the resulting nonlinear equations. The analytical method developed allows combining these two stages.

The problem under consideration has some useful features that can be utilized when constructing an approximate analytical method. In particular, they include a simple definitional domain of the multidimensional integral (namely, a parallelepiped) and the subintegral functions represented by exponentials of continuous functions.

Note that the exponential function is analytic, and the Lagrange multipliers and the model output are bounded. Therefore, we have the following polynomial approximation of degree q :

$$\exp(-v_{(n)}) = \sum_{h=0}^q \frac{(-1)^h}{h!} v_{(n)}^h, \quad (15)$$

where

$$v_{(n)} = \langle \boldsymbol{\lambda}^{(n)}, \mathbf{z}^{(n)}(\mathbf{a}) \rangle < \pm M < \infty \quad (16)$$

and³

$$(v_{(n)})^h = \sum_{i_j \geq 0; \sum_{j=1}^s i_j = h} A_{i_1, \dots, i_h}^{(h)} (\lambda_1^{(n)})^{i_1} \dots (\lambda_s^{(n)})^{i_h} (z^1[n | \mathbf{a}])^{i_1} \dots (z^s[n | \mathbf{a}])^{i_h}, \quad (17)$$

$$A_{i_1, \dots, i_h}^{(h)} = \frac{h!}{i_1! \dots i_h!}. \quad (18)$$

With this notation, system (14) takes the form

$$\sum_{h=0}^q \frac{(-1)^h}{h!} \sum_{(i_1, \dots, i_h)=1}^s \lambda_{i_1}^{(n)} \dots \lambda_{i_h}^{(n)} u_{i_1, \dots, i_h; k}^{(n)} - v_n^k = 0, \quad (19)$$

where

$$u_{i_1, \dots, i_h; k}^{(n)} = \int_{\mathcal{A}} z^{i_1}[n | \mathbf{a}] \dots z^{i_h}[n | \mathbf{a}] \left(z^k[n | \mathbf{a}] - y^{(k)}[n] \right) d\mathbf{a}, \quad i_1, \dots, i_h, \quad k = \overline{1, s}. \quad (20)$$

The numbers $i_1, \dots, i_h$ take values in the interval $[1, s]$. This system contains s variables (the Lagrange multipliers), and each equation is a multivariate polynomial of degree q .

By assumption, the model is characterized by a continuous function (or functional). Hence, it can be represented by a multivariate polynomial, and then the multidimensional integrals in (20) are transformed into the product of one-dimensional integrals calculated analytically. Therefore, the balance system (19) has a polynomial left-hand side.

Consider the following family of systems in the parameter $\varepsilon \in [0, 1]$:

$$- \sum_{i_1=1}^s \lambda_{i_1}^{(n)} u_{i_1, k}^{(n)} + \varepsilon \sum_{h=1}^q \frac{(-1)^h}{h!} \sum_{\substack{(i_1, \dots, i_h)=1 \\ \sum_{j=1}^h i_j = h}}^s \lambda_{i_1}^{(n)} \dots \lambda_{i_h}^{(n)} u_{i_2, \dots, i_h; k}^{(n)} - v_n^k = 0. \quad (21)$$

For $\varepsilon = 0$, we have the so-called *basic* system, linear with a square matrix $U^{(n)} = [u_{i_1, k}^{(n)} | (i_1, k) = \overline{1, s}]$:

$$U^{(n)} \boldsymbol{\lambda}^{(n)} = -\mathbf{v}_n, \quad (22)$$

where $\mathbf{v}_n = \{v_n^1, \dots, v_n^s\}$.

If $\det U^{(n)} \neq 0$, then the Lagrange multipliers yielding the *basic* solution are

$$\boldsymbol{\lambda}_{(\bullet)}^{(n)} = -[U^{(n)}]^{-1} \mathbf{v}_n. \quad (23)$$

We write the solution of system (21) as an abstract power series in the parameter ε [23, 24]:

$$\lambda_{k, \star}^{(n)} = \lambda_{k, \bullet}^{(n)} + \varepsilon \lambda_{k, I}^{(n)} + \varepsilon^2 \lambda_{k, II}^{(n)} + \dots, \quad k = \overline{1, s}, \quad (24)$$

where $\lambda_{k, I}^{(n)}, \lambda_{k, II}^{(n)}, \dots$ are the first, second, ..., corrections to the basic solution, respectively.

To find the corrections sequentially, we apply the method of undetermined coefficients [23] to obtain:

— the first correction vector

$$*\boldsymbol{\lambda}_I^{(n)} = -[U^{(n)}]^{-1} \mathbf{b}_{(1)}^{(n)}(\boldsymbol{\lambda}_{(\bullet)}^{(n)}), \quad (25)$$

where

$$\mathbf{b}_{(I)}^{(n)}(\boldsymbol{\lambda}_{(\bullet)}^{(n)}) = \left\{ \frac{1}{2} \sum_{(i_1, i_2)=1}^s \lambda_{i_1, \bullet}^{(n)} \lambda_{i_2, \bullet}^{(n)} u_{i_1, i_2, 1}^{(n)}, \dots, \frac{1}{2} \sum_{(i_1, i_2)=1}^s \lambda_{i_1, \bullet}^{(n)} \lambda_{i_2, \bullet}^{(n)} u_{i_1, i_2, s}^{(n)} \right\}, \quad (26)$$

³ The expression on the right-hand side of (17) is Newton's polynomials of degree h [31].

and the second correction vector

$$^*\lambda_{(II)}^n = -[U^{(n)}]^{-1} \mathbf{b}_{(II)}^{(n)}(\lambda_{(\bullet)}^n, \lambda_{(I)}^n), \quad (27)$$

where

$$\left\{ \frac{1}{2} \sum_{(i_1, i_2)=1}^s \lambda_{i_1, I}^{(n)} \lambda_{i_2, \bullet}^{(n)} u_{i_1, i_2, k}^{(n)} + \frac{1}{3!} \sum_{(i_1, i_2, i_3)=1}^s \lambda_{i_1, \bullet}^{(n)} \lambda_{i_2, \bullet}^{(n)} \lambda_{i_3, \bullet}^{(n)} u_{i_1, i_2, i_3, k}^{(n)} \right\}. \quad (28)$$

Thus, the solution of the balance system (21) is written as

$$\lambda_{\star}^{(n)} = \lambda_{(\bullet)}^{(n)} + ^*\lambda_I^{(n)} + ^*\lambda_{II}^{(n)} + \dots. \quad (29)$$

3.3. Asset Pricing Model

Consider the pricing process of an asset during trading sessions and apply the above method for generating random data with given numerical characteristics to forecast price dynamics.

By a theoretical consensus, the price of an asset at any time instant is the product of balancing real demand and real supply. However, a trader is guided by the expected demand and supply, which may significantly differ from the real ones. In these conditions, price forecasting under sufficiently high uncertainty becomes crucial.

Therefore, it seems natural to attempt to maximize the information entropy as a measure of uncertainty on training retrospective data containing the values of mean price and its second moment (variance describing mean volatility).

3.3.1. Price Dynamics Model

We adopt an autonomous model in the form of a linear difference equation of order p , but with random interval-type parameters:

$$C[t] = \sum_{i=1}^p a_i C[t-i], \quad t \in \mathcal{T}, \quad a_i \in \mathcal{A}_i = [d, w], \quad \mathcal{A}^p = \prod_{i=1}^p \mathcal{A}_i. \quad (30)$$

(All parameters have the same intervals $[d, w]$.)

We will use this model at the stages of *training*, $\mathcal{T}_{trn} = [t_0, t_0 + p]$, and *forecasting*, $\mathcal{T}_{frc} = [t_0 + p + 1, t_0 + p + 1 + t_{frc}]$, where t_{frc} are one- or two-day forecasts. (Longer forecasts can be considered by analogy.)

The probabilistic properties of the parameters are characterized by a continuously differentiable PDF $P_t(\mathbf{a})$.

By assumption, based on the results of each real trading session t , two price indicators are formed: the mean price $m_C^*[t]$ and its second moment $D_C^*[t]$ as a characteristic of mean volatility:

$$D_C^*[t] = (V_C^*[t])^2 + (m_C^*[t])^2, \quad (31)$$

where $V_C^*[t]$ is the estimated standard deviation of the price, constructed from the value of its maximum and minimum deviation.

3.3.2. Data

Consider real price and volatility dynamics data available on a past interval $[t_0 - p, t_0 - 1]$:

$$m_C^*[t_0 - p], m_C^*[t_0 - p + 1], \dots, m_C^*[t_0 - 1]$$

and

$$D_C^*[t_0 - p], D_C^*[t_0 - p + 1], \dots, D_C^*[t_0 - 1],$$

respectively.

Using these data, the model (30) generates the prices on the training interval $\mathcal{T}_{trn}$:

$$\mathbb{C}[\mathbf{a} | t] = \sum_{i=1}^p a_i m_C^*[t - i], \quad t \in \mathcal{T}_{trn}. \quad (32)$$

3.3.3. The Entropy-Optimal Estimation of the PDFs of Model Parameters

To optimize the PDF $P_t(\mathbf{a})$, we will apply randomized machine learning [22], see formulas (9)–(11). This methodology is reduced to solving the following problems for each trading session $t \in \mathcal{T}_{trn}$:

$$\mathcal{H}_t[P_t(\mathbf{a})] = - \int_{\mathcal{A}^p} P_t(\mathbf{a}) \ln P_t(\mathbf{a}) d\mathbf{a} \Rightarrow \max_{P(\mathbf{a})} \quad (33)$$

subject to the constraints

$$\int_{\mathcal{A}^p} P_t(\mathbf{a}) d\mathbf{a} = 1, \quad (34)$$

$$\int_{\mathcal{A}^p} P_t(\mathbf{a}) \mathbb{C}[\mathbf{a} | t] d\mathbf{a} = m_C^*[t], \quad \int_{\mathcal{A}} P_t(\mathbf{a}) \mathbb{C}^2[\mathbf{a} | t] d\mathbf{a} = D_C^*[t], \quad t \in \mathcal{T}_{trn}, \quad (35)$$

where $\mathbb{C}_t[\mathbf{a} | t]$ are given by (32).

According to (12) and (13), problem (33)–(35) has the analytical solution

$$P_t^*(\mathbf{a}) = \frac{\exp(-\lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t] - \lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t])}{\mathcal{P}_t(\lambda_1^{(t)}, \lambda_2^{(t)})}, \quad t \in \mathcal{T}_{trn}. \quad (36)$$

$$\mathcal{P}_t^*(\lambda_1^{(t)}, \lambda_2^{(t)}) = \int_{\mathcal{A}^p} \exp(-\lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t] - \lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t]) d\mathbf{a},$$

The Lagrange multipliers $\lambda_1^{(t)}, \lambda_2^{(t)}$ are found by solving the two balance equations

$$\begin{aligned} \int_{\mathcal{A}^p} \exp(-\lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t] - \lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t]) (\mathbb{C}[\mathbf{a} | t - 1] - m_C^*[t]) d\mathbf{a} &= 0, \\ \int_{\mathcal{A}^p} \exp(-\lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t] - \lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t]) (\mathbb{C}^2[\mathbf{a} | t - 1] - D_C^*[t]) d\mathbf{a} &= 0, \end{aligned} \quad t \in \mathcal{T}_{trn}. \quad (37)$$

According to the method developed in [29], we approximate the exponent by a polynomial of degree 2 :

$$\begin{aligned} \exp(x) &\approx \left(1 - \lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t] + \frac{1}{2} \left(\lambda_1^{(t)} \mathbb{C}[\mathbf{a} | t]\right)^2\right) \mathbb{C}[\mathbf{a} | t], \\ \exp(y) &\approx \left(1 - \lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t] + \frac{1}{2} \left(\lambda_2^{(t)} \mathbb{C}^2[\mathbf{a} | t]\right)^2\right) \mathbb{C}^2[\mathbf{a} | t]. \end{aligned} \quad (38)$$

With the approximations (38), the balance equations take the form

$$\begin{aligned} \lambda_1 B_1(t) + \lambda_2 B_2(t) + \lambda_1^2 B_3(t) + \lambda_2^2 B_4(t) + \lambda_1 \lambda_2 B_5(t) &= B_0(t), \\ \lambda_1 Z_1(t) + \lambda_2 Z_2(t) + \lambda_1^2 Z_3(t) + \lambda_2^2 Z_4(t) + \lambda_1 \lambda_2 Z_5(t) &= Z_0(t), \end{aligned} \quad t \in \mathcal{T} = [t_0, t_0 + p]. \quad (39)$$

In the first equation above, the coefficients are

$$\begin{aligned} B_0(t) &= A m_C^*[t] - I_{p,1}^{(t)}, \quad B_1(t) = m_C^*[t] I_{p,1}^{(t)} - I_{p,2}^{(t)}, \\ B_2(t) &= m_C^*[t] I_{p,2}^{(t)} - I_{p,3}^{(t)}, \quad B_3(t) = -\frac{1}{2} B_2^{(2,3)}(t), \\ B_4(t) &= -\frac{1}{2} \left(m_C^*[t] I_{p,4}^{(t)} - I_{p,5}^{(t)}\right), \quad B_5(t) = -\left(m_C^*[t] I_{p,3}^{(t)} - I_{p,4}^{(t)}\right). \end{aligned} \quad (40)$$

In the second equation, the coefficients are

$$\begin{aligned} Z_0(t) &= A D_C^*[t] - I_{p,2}^{(t)}, \quad Z_1(t) = D_C^*[t] I_{p,1}^{(t)} - I_{p,3}^{(t)}, \\ Z_2(t) &= D_C^*[t] I_{p,2}^{(t)} - I_{p,4}^{(t)}, \quad Z_3(t) = -\frac{1}{2} Z_2^{(2,4)}(t), \\ Z_4(t) &= -\frac{1}{2} \left(D_C^*[t] I_{p,4}^{(t)} - I_{p,6}^{(t)} \right), \quad Z_5(t) = -\left(D_C^*[t] I_{p,3}^{(t)} - I_{p,5}^{(t)} \right). \end{aligned} \quad (41)$$

In these expressions,

$$\begin{aligned} A &= \int_{\mathcal{A}} d\mathbf{a} = (w-d)^p, \\ I_{p,n}^{(t)}(k_1, \dots, k_n) &= \underbrace{\int_d^w \dots \int_d^w}_{p} \mathbb{C}^n[\mathbf{a} | t] d\mathbf{a} \\ &= \sum_{k_j \geq 0; \sum_{j=1}^n k_j = n} \frac{n!}{k_1! \dots k_n!} \left(\underbrace{\int_d^w \dots \int_d^w}_{p} a_1^{k_1} \dots a_p^{k_n} da_1 \dots da_p \right) \\ &\quad \times (m_C^*[t])^{k_1} \dots (m_C^*[t-p])^{k_n}, \quad n = \overline{0, 6}. \end{aligned} \quad (42)$$

4. FORECASTING THE FUTURE PRICE OF A TRADED ASSET

We will experimentally test the method for the *one- and two-day forecasting* of the mean price and mean volatility of a traded asset, i.e., Gazprom's stocks during 2020 on the Moscow Exchange.

Consider twelve trading sessions, each at the beginning of the month. For the convenience of further calculations, let us introduce a conventional monetary unit (c.m.u. = 1000 rubles). The stock price data in c.m.u. are combined in Table 1.

Table 1. Gazprom's stock price quotations in 2020

Month	Jan.	Feb.	Mar.	Apr.	May	Jun.
Data t	1	2	3	4	5	6
Price m_C^*	0.259	0.223	0.208	0.178	0.188	0.200
Max $C_{\max}^*$	0.262	0.240	0.212	0.196	0.202	0.208
Min $C_{\min}^*$	0.227	0.201	0.158	0.177	0.182	0.190
V_C^*	0.017	0.020	0.027	0.009	0.011	0.009
D_C^*	0.084	0.069	0.070	0.041	0.046	0.049
Month	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
Date t	7	8	9	10	11	12
Price m_C^*	0.195	0.182	0.181	0.171	0.154	0.183
Max $C_{\max}^*$	0.202	0.195	0.186	0.173	0.189	0.215
Min $C_{\min}^*$	0.179	0.180	0.170	0.154	0.152	0.182
V_C^*	0.011	0.007	0.009	0.010	0.019	0.017
D_C^*	0.206	0.189	0.190	0.181	0.173	0.200

4.1. Training of the Price Model

The price model (30) possesses a memory of $p = 2$, the initial training instant is $t_0 = 3 \rightarrow \text{Mar.}$, and the interval limits of the two parameters are $d = -1$, $w = 2$.

The price model will be trained on the interval $\mathcal{T} = [t_0, t_0 + 2] = [3, 5]$. The historical period is $\mathcal{I}_p = [t_0 - 2, t_0 - 1] = [1, 2]$.

Since model (32) contains the two parameters,

$$\mathbb{C}[\mathbf{a} | t] = a_1 m_C^*[t - 1] + a_2 m_C^*[t - 2]. \quad (43)$$

The PDFs in the corresponding trading sessions have the form (36) with model (43). Note that the entropy-optimal PDFs of the parameters generated by the linear model (43) differ from the Gaussian distribution.

To find the Lagrange multipliers (solve the balance equations), we apply the approximate analytical method from subsection 2.2.

For trading sessions $t = 3, 4, 5$, the entropy-optimal PDFs with the approximate Lagrange multipliers within the first correction have the form

$$P_3^*(\mathbf{a} | 1.068; -0.871) = 0.131 \exp(-0.238a_1 - 0.277a_2 + 0.043a_1^2 + 0.058a_2^2 + 0.100a_1a_2), \quad (44)$$

$$P_4^*(\mathbf{a} | 0.958; 0.102) = 0.133 \exp(-0.199a_1 - 0.214a_2 - 0.004a_1^2 - 0.005a_2^2 - 0.005a_1a_2), \quad (45)$$

$$P_5^*(\mathbf{a} | -1.994; 2.609) = 0.092 (0.355a_1 + 0.415a_2 - 0.083a_1^2 - 0.112a_2^2 - 0.193a_1a_2). \quad (46)$$

Their plots are shown in Figs. 1–3.

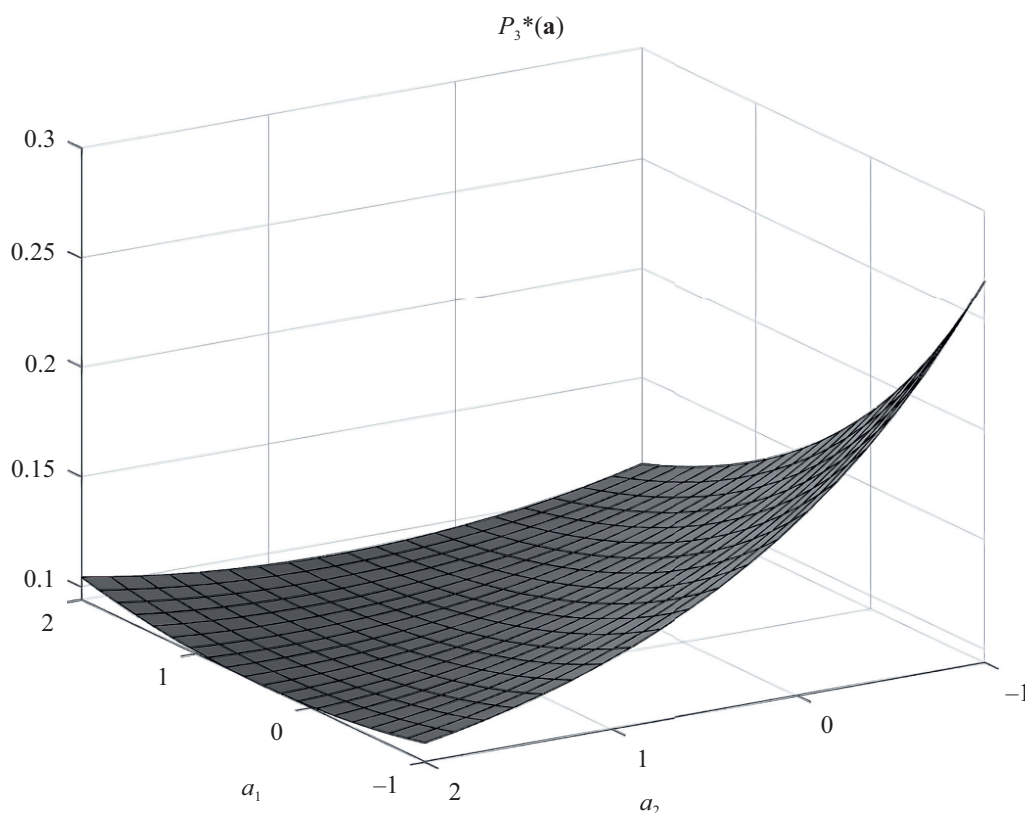


Fig. 1. The distribution $P_3^*(\mathbf{a})$.

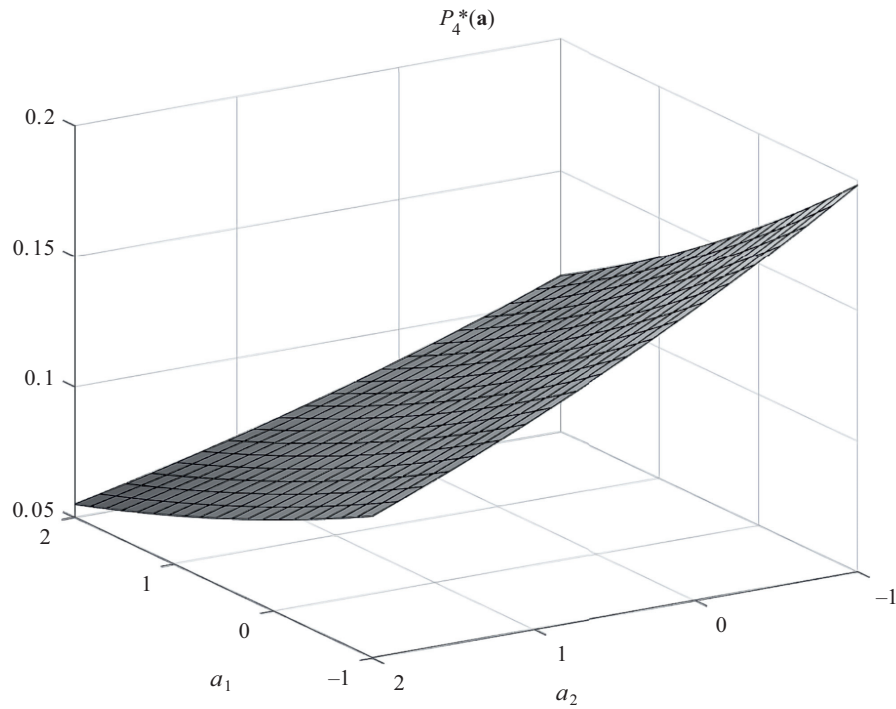


Fig. 2. The distribution $P_4^*(\mathbf{a})$.

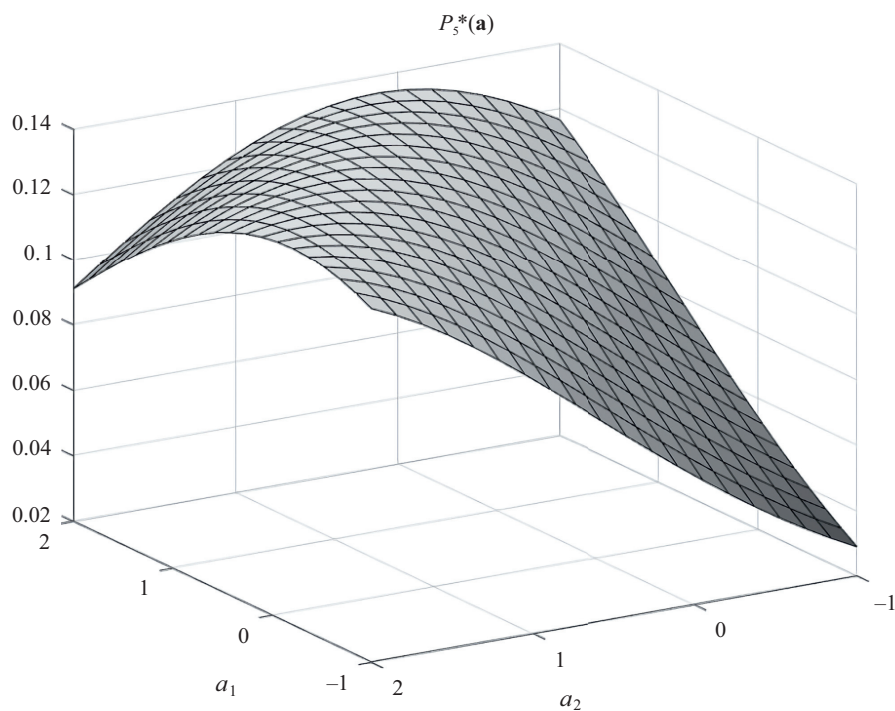


Fig. 3. The distribution $P_5^*(\mathbf{a})$.

4.2. Forecasting the Mean Price and Mean Volatility

The above entropy-optimal PDFs $P_3^*(\mathbf{a})$, $P_4^*(\mathbf{a})$, and $P_5^*(\mathbf{a})$ of the model parameters (32) will be used to generate data ensembles and then calculate the forecasted values of $m_C[t]$ and $D_C[t]$ in trading sessions from Apr. (4) to Nov. (11). The realized values of these variables are known (Table 1); hence, it is possible to estimate the accuracy of different forecasting strategies.

4.2.1. One-Day Forecasts $P_k^*(\mathbf{a}) \rightarrow (m_C[k+1], V_C[k+1])$

For one-day forecasts, we use the optimal PDF for trading session k to predict the results of trading session $(k+1)$. Consider the procedure for constructing the forecast $3 \rightarrow 4$. For this purpose, it is necessary to use the PDF $P_3^*(\mathbf{a})$ (44) and the forecasting model (32); in this example, the model takes the form

$$\mathbb{C}[\mathbf{a} | 4] = a_1 m_C^*[3] + a_2 m_C^*[2]. \quad (47)$$

We transform the PDF $P_3^*(\mathbf{a})$ (44) into the random sequence $\{a_1, a_2\}$. The generated ensemble contains 1000 values $\mathbb{C}[\mathbf{a} | 4]$. We calculate $\bar{m}_C[4] = \tilde{\mathcal{M}}(\mathbb{C}[\mathbf{a} | 4])$ and $\bar{\sigma}_C^2[4] = \tilde{\mathcal{M}}\{(\mathbb{C}[\mathbf{a} | 4] - \bar{m}_C[4])^2\}$, where $\tilde{\mathcal{M}}\{\bullet\}$ denotes the empirical mean operator.

The forecasts $4 \rightarrow 5$ and $5 \rightarrow 6$ are constructed by analogy. Table 2 presents the resulting one-day forecasts and their accuracy estimates compared to the realized values in the trading sessions.

Table 2. One-day forecasts

Forecast $\bullet$	$3 \rightarrow 4$	$4 \rightarrow 5$	$5 \rightarrow 6$
$\bar{m}_C[\bullet]$	0.175	0.145	0.229
$m_C^*[\bullet]$	0.178	0.188	0.200
$\bar{\sigma}_C^2[\bullet]$	0.076	0.056	0.041
$V_C^*[\bullet]$	0.041	0.046	0.049
$ \delta_m[\bullet] $	0.003	0.043	0.029
$ \delta_\sigma[\bullet] $	0.035	0.010	0.008

In this table, the variables are

$$\delta_m[\bullet] = \bar{m}_C[\bullet] - m_C^*[\bullet], \quad \delta_\sigma[\bullet] = \bar{\sigma}_C^2[\bullet] - V_C^*[\bullet]. \quad (48)$$

Figures 4–6 show the empirical PDFs of the forecasted prices in trading sessions 4, 5, and 6 under one-day forecasting.

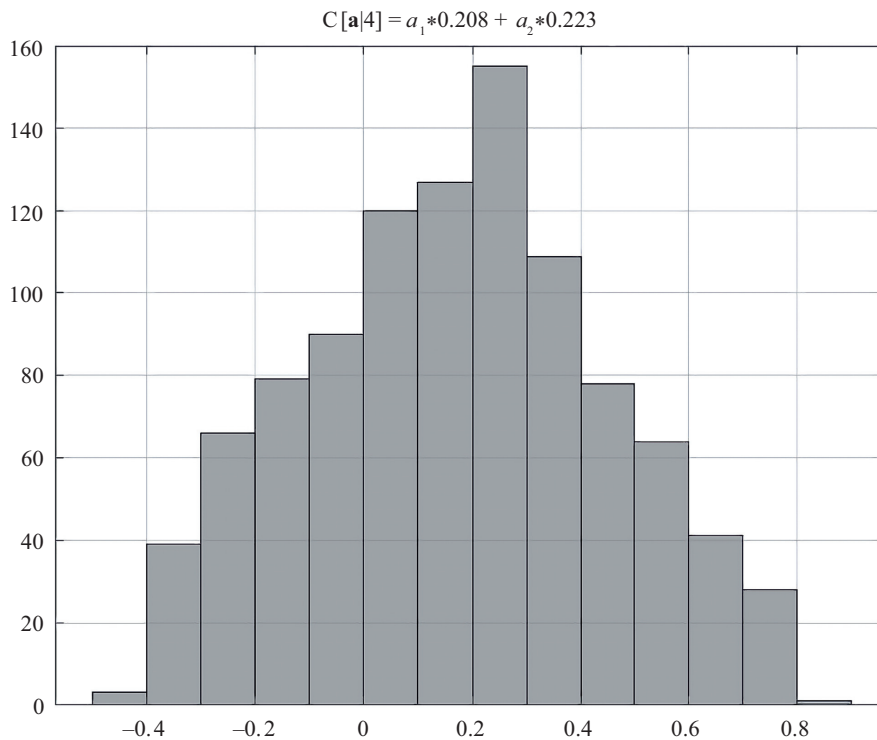


Fig. 4. The empirical PDF $\mathbb{C}[\mathbf{a} | 4]$ (one-day forecast $3 \rightarrow 4$).

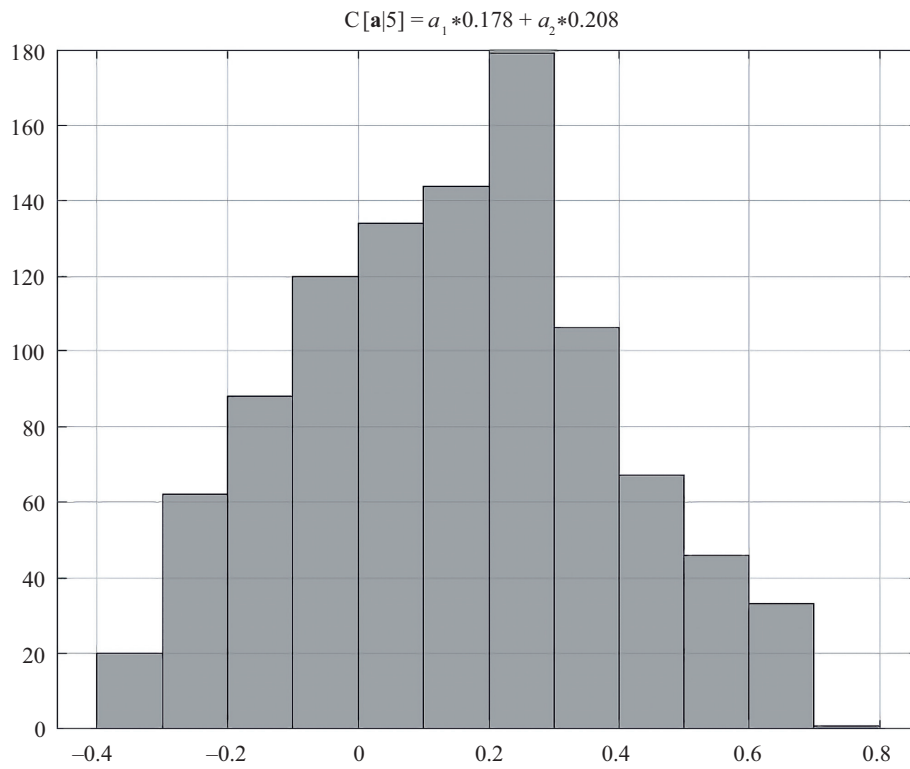


Fig. 5. The empirical PDF $C[a|5]$ (one-day forecast $4 \rightarrow 5$).

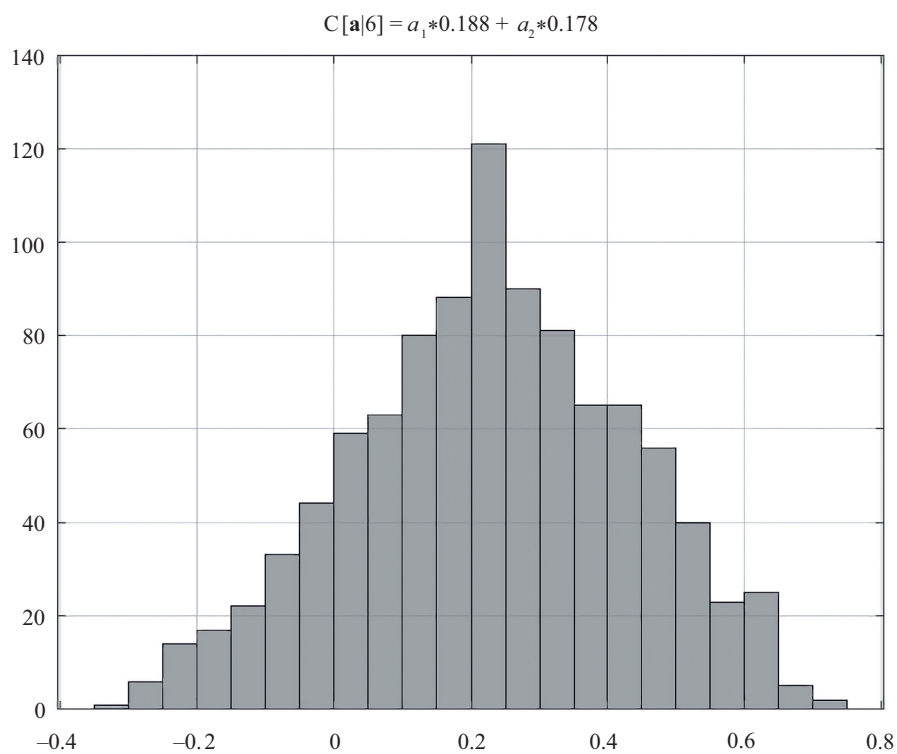


Fig. 6. The empirical PDF $C[a|6]$ (one-day forecast $5 \rightarrow 6$).

The integral relative error of the *mean price forecast* under one-day forecasting is given by

$$\Delta_m = \frac{\sqrt{\sum_{t=4}^6 \delta_m^2[t]}}{\sqrt{\sum_{t=4}^6 m_C^2[t] + \sum_{t=4}^6 (m_C^*[t])^2}} = 8\%. \quad (49)$$

The integral relative error of the *mean volatility forecast* under one-day forecasting is given by

$$\Delta_\sigma = \frac{\sqrt{\sum_{t=4}^6 \delta_\sigma^2[t]}}{\sqrt{\sum_{t=4}^6 \sigma_C^2[t] + \sum_{t=4}^6 (V_C^*[t])^2}} = 17\%. \quad (50)$$

4.2.2. Two-Day Forecasts $P_k^*(\mathbf{a}) \rightarrow (m_C[k+1], V_C[k+1]), (m_C[k+2], V_C[k+2])$

In two-day forecasts: the optimal PDF for trading session k is used to predict the results of trading session $(k+2)$.

The forecast $3 \rightarrow 4, 5$ with the PDF $P_3^*(\mathbf{a})$ can be implemented as

$$\begin{aligned} \mathbb{C}[\mathbf{a} | 4] &= a_1 m^*[3] + a_2 m_C^*[2], \\ \mathbb{C}[\mathbf{a} | 5] &= a_1 m^*[4] + a_2 m_C^*[3] \end{aligned} \quad (51)$$

(sequential one-day forecasts) or as

$$\begin{aligned} \mathbb{C}[\mathbf{a} | 4] &= a_1 m^*[3] + a_2 m_C^*[2] = a_1 0.208 + a_2 0.223, \\ \mathbb{C}[\mathbf{a} | 5] &= a_1 \bar{m}^*[4] + a_2 m_C^*[3] = a_1 \bar{m}^*[4] + a_2 0.208, \\ \bar{m}^*[4] &= \tilde{\mathcal{M}}\{\mathbb{C}[\mathbf{a} | 4]\} \end{aligned} \quad (52)$$

(using information from the first one-day forecast).

The two-day forecasts $4 \rightarrow 5, 6$ and $5 \rightarrow 6, 7$ are constructed by analogy. Table 3 presents the resulting two-day forecasts and their accuracy estimates compared to the realized values in the trading sessions.

Table 3. Two-day forecasts

•	$3 \rightarrow 4$	$3 \rightarrow 5$	$4 \rightarrow 5$	$4 \rightarrow 6$	$5 \rightarrow 6$	$5 \rightarrow 7$
$\bar{m}_C[\bullet]$	0.175	0.185	0.145	0.206	0.229	0.200
$m_C^*[\bullet]$	0.178	0.188	0.188	0.200	0.200	0.195
$\bar{\sigma}_C^2[\bullet]$	0.056	0.060	0.056	0.012	0.041	0.024
$V_C^*[\bullet]$	0.041	0.051	0.046	0.009	0.049	0.011
$ \delta_m[\bullet] $	0.003	0.033	0.043	0.078	0.029	0.065
$ \delta_\sigma[\bullet] $	0.035	0.049	0.010	0.030	0.009	0.043

The integral relative error of the *mean price forecast* under two-day forecasting is given by

$$\Delta_m = \frac{\sqrt{\sum_{t=5}^7 \delta_m^2[t]}}{\sqrt{\sum_{t=5}^7 m_C^2[t] + \sum_{t=5}^7 (m_C^*[t])^2}} = 7.2\%. \quad (53)$$

The integral relative error of the *mean volatility forecast* under two-day forecasting is given by

$$\Delta_\sigma = \frac{\sqrt{\sum_{t=5}^7 \delta_\sigma^2[t]}}{\sqrt{\sum_{t=5}^7 \sigma_C^2[t] + \sum_{t=5}^7 (V_C^*[t])^2}} = 25\%. \quad (54)$$

Figures 7–9 show the empirical PDFs of the forecasted prices in trading sessions 5–7 under two-day forecasting.

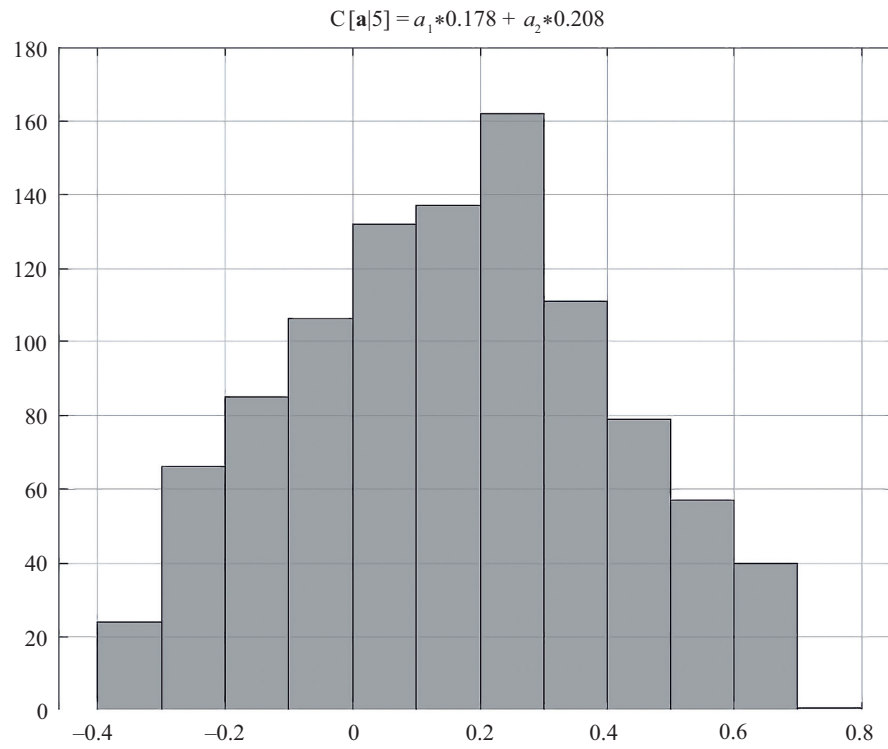


Fig. 7. The empirical PDF $C[a|5]$ (two-day forecast $3 \rightarrow 4, 5$).

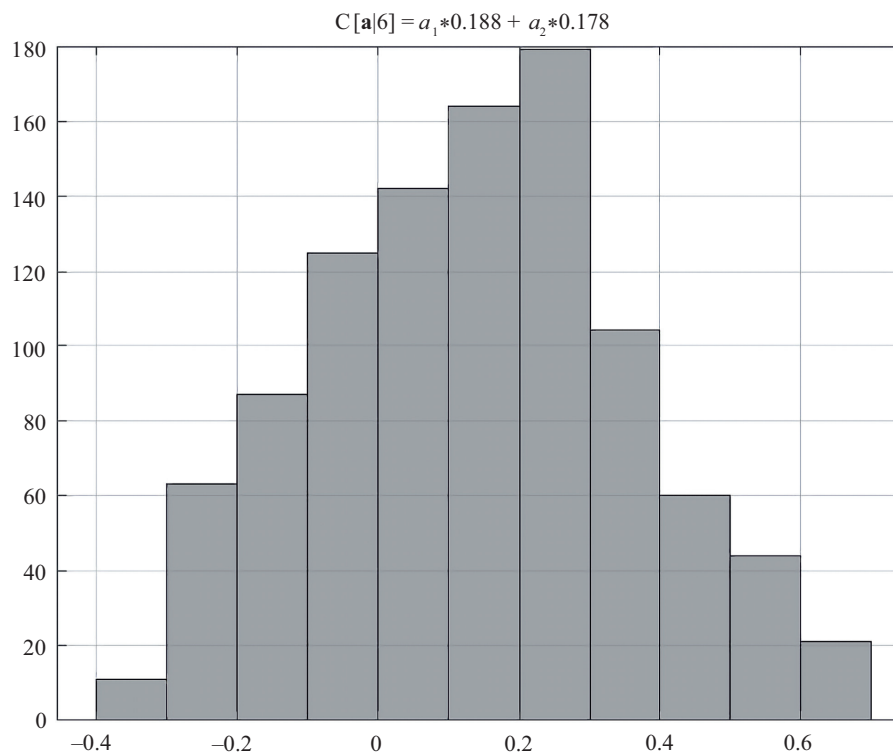


Fig. 8. The empirical PDF $C[a|6]$ (two-day forecast $4 \rightarrow 5, 6$).

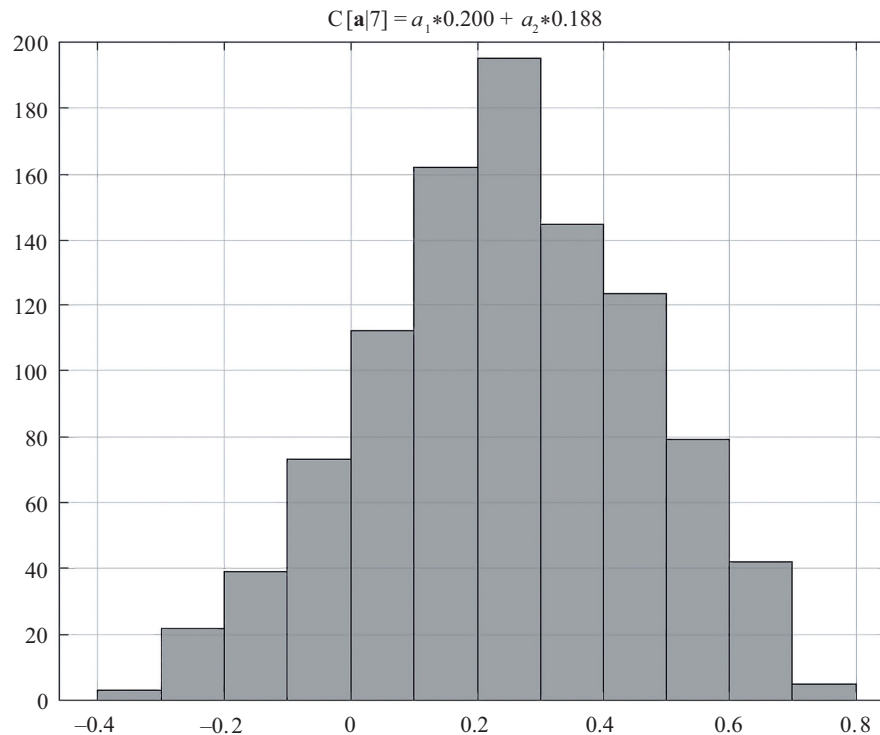


Fig. 9. The empirical PDF $C[a|7]$ (two-day forecast $5 \rightarrow 6, 7$)

5. DISCUSSION

The problem of generating suitable data for testing and forecasting is quite popular in modern computer science. In this paper, we have adapted and further developed the technology of randomized machine learning for generating data ensembles with given numerical characteristics.

Unlike the existing technology, an extension has been proposed to consider moment characteristics from the 1st to s th order. According to the results, this approach leads to non-Gaussian PDFs even in the case of a linear data model. Similar to the existing technology, the extension is reduced to solving the corresponding balance equations with integral components. The paper has presented an approximate analytical solution method for these equations based on power series and the method of undetermined coefficients.

It has been applied to forecast the price of an asset, and the results have been compared with the realized data for one- and two-day forecasts. Within this study, quite acceptable accuracy of the approximate solution has been discovered through two corrections. However, in-depth research into the approximate method is necessary, both in its theoretical aspects and numerical simulation.

6. CONCLUSIONS

This paper has presented a theory and algorithm for generating test data ensembles with specified properties (numerical characteristics) based on a structural modification of the randomized machine learning procedure [22]. As is known, the core of this procedure is the balance equations for the Lagrange multipliers, which contain the so-called integral components (the multidimensional integrals of arbitrary subintegral functions).

The *analytical* solution method developed in [29] has been adapted for solving these equations. With this method, the multidimensional integration problem is reduced to calculating the sum of products of the one-dimensional integrals of power functions.

Finally, a randomized forecasting method has been developed and applied to construct one- and two-day forecasts of the mean price and mean volatility of a traded asset.

FUNDING

This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. 075-15-2024-544.

REFERENCES

1. Rubinstein, R.Y. and Kroese, D.P., *Simulation and the Monte Carlo Method*, John Wiley & Sons, 2016.
2. Vapnik, V.N., *Statistical Learning Theory*, Wiley, 1998.
3. Bishop, C.M., *Pattern Recognition and Machine Learning*, Springer, 2006.
4. Hastie, T., Tibshirant, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, 2009.
5. Vovk, V. and Shafer, G., Good Randomized Sequential Probability Forecasting Is Always Possible, *Journal of Royal Statistical Society B*, 2005, vol. 67, no. 5, pp. 747–763.
6. Hong, T., Prinson, P., Fan, S., Zareijpour, H., Triccoli, A., and Hyndman, R.J., Probabilistic Energy Forecasting: Global Energy Forecasting Competition 2014 and Beyond, *Inter. Journal of Forecasting*, 2016, vol. 32, no. 3, pp. 896–913.
7. Zhang, L., Aggarwal, C.C., and Qi, G.-J., Stock Price Prediction via Discovering Multy-Frequency Trading Patterns, *Proc. of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2017, pp. 2141–2149.
8. Myers, G.J., *The Art of Software Testing*, John Wiley & Sons, 1979.
9. Gorodetsky, V.I., Grushitsky, M.S., and Khabalov, A.V., Multiagent Systems, *Artificial Intelligence News*, 1998, no. 2, pp. 64–116.
10. Patton, R., *Software Testing*, SAWS Publishers, 2005.
11. Lysikov, A.I., But, G.S., and Didenko, D.A., A Computer Test Development System, 2002. <http://www.bytic.ru/cue99m/cf7pvke.html>
12. Micel, A.A. and Poguda, A.A., Neural Networks Model Approach to Knowledge Testing, *Applied Informatics*, 2011, no. 5(35), pp. 60–67.
13. Zaozerskaya, L.A. and Platonova, V.A., Mathematical Models to Form an Optimal Set of Test Structures for Knowledge Control, *Omsk Scientific Bulletin*, 2012, no. 3, pp. 33–36.
14. Campi, M.C., Garatti, S., and Prandini, M., The Scenario Approach for Systems and Control Design, *Ann. Rev. Control*, 2009, vol. 33, no. 2, pp. 149–157.
15. Chi, Z., Liu, Y., Turrini, A., Zhang, L., and Jansen, D.N., A Scenario Approach for Parametric Markov Decision Processes, in *Principles of Verification: Cycling the Probabilistic Landscape: Essay Dedicated to Joost-Pieter Katoen on the Occasion of His 60th Birthday, Part II*, Cham: Springer, 2024, pp. 234–266.
16. Boltzmann, L., *Vorlesungen uber Gastheory*, Leipzig, 1896, vol. 1, J.A. Barth; 1898, vol. 2, J.A. Barth.
17. Jaynes, E.T., Information Theory and Statistical Mechanics, *Physical Review*, 1957, vol. 106, no. 4, pp. 620–630.
18. Jaynes, E.T., Gibbs vs. Boltzmann Entropy, *American Journal of Physics*, 1965, vol. 33, pp. 391–398.
19. Rosenkrantz, R.D., *E.T. Jaynes: Papers on Probability, Statistics, and Statistical Physics*, Kluwer Academic Publishers, 1989.
20. Jaynes, E.T., *Probability Theory: the Logic of Science*, Cambridge: Cambridge Univ. Press, 2003.

21. Popkov, Y.S., Asymptotic Efficiency of Maximum Entropy Estimates, *Dokl. Math.*, 2020, vol. 102, pp. 350–352.
22. Popkov, Yu.S., Popkov, A.Yu., and Dubnov, Yu.A., *Entropy Randomization in Machine Learning*, CRC Press, 2023.
23. Krasnosel'skii, M.A., Vainikko, G.M., Zabreiko, P.P., Rutitski, Ja.B., and Stecenko, V.Ja., *Approximated Solutions of Operator Equations*, Groningen: Walters-Noordhoff, 1972.
24. Malkin, I.G., *Some Problems in the Theory of Nonlinear Oscillation*, U.S. Atomic Energy Commission, Technical Information Service, 1959.
25. Darkhovsky, B.S., Popkov, Y.S., Popkov, A.Y., and Aliev, A.S., A Method of Generating Random Vectors with a Given Probability Density Function, *Autom. Remote Control*, 2018, vol. 79, no. 9, pp. 1569–1581. <https://doi.org/10.1134/S0005117918090035>
26. Avellaneda, M., Minimum-Relative-Entropy Calibration of Asset-Pricing Models, *International Journal of Theoretical and Applied Finance*, 1998, vol. 1, no. 04, pp. 447–472.
27. Jackwerth, J.C., Recovering Risk Aversion from Option Prices and Realized Returns, *The Review of Financial Studies*, 2000, vol. 13, no. 2, pp. 433–451.
28. Ant-Sahalia, Y. and Lo, A.W., Nonparametric Risk Management and Implied Risk Aversion, *Journal of Economics*, 2000, vol. 94, no. 1–2, pp. 9–51.
29. Popkov, Yu.S., Analytic Method for Solving One Class of Nonlinear Equations, *Dokl. Math.*, 2024, vol. 110, pp. 404–407. <https://doi.org/10.1134/S1064562424601392>
30. Fikhtengol'ts, G.M., *Kurs differentsial'nogo i integral'nogo ischisleniya* (Course on Differential and Integral Calculus), Moscow: Fizmatgiz, 1962.
31. Feller, W., *An Introduction to Probability Theory and Its Applications*, Wiley, 1968.
32. Sobol, I.M., *The Monte Carlo Method*, Moscow: Mir Publishers, 1975.
33. Bakhvalov, N.S., Zhidkov, N.P., and Kobel'kov, G.M., *Chislennyye metody* (Numerical Methods), Moscow: Binom, 2003.

This paper was recommended for publication by O.N. Granichin, a member of the Editorial Board

On the Emergence of Binary Mapping Systems

A. N. Karkishchenko

*Research and Design Bureau of Robotics and Control Systems,
Southern Federal University, Taganrog, Russia
e-mail: karkishalex@gmail.com*

Received February 2, 2025

Revised March 28, 2025

Accepted April 17, 2025

Abstract—The emergence effect is considered in terms of mappings realized by abstract systems that describe observable physical manifestations. The emergence phenomenon is interpreted in terms of the expansion of the class of realized mappings when elements are combined into a system or new connections between elements arise. The paper considers the problem of describing emergent properties in models of systems that realize binary mappings. The structure of interaction of elements in such systems is described by a finite oriented graph. Classes of mappings are studied for set-theoretical operations on such systems. A “superadditive” expansion of the class of mappings is shown when systems are combined. The emergence coefficient is introduced and substantiated. Lower and upper bounds for this coefficient are proved.

Keywords: binary mapping, system, set-theoretic operations, emergence, emergence coefficient

DOI: 10.31857/S0005117925070076

1. INTRODUCTION

In systems analysis, the concept of emergence is one of the most interesting properties of complex systems. In the standard formulation, emergence is the appearance of new properties and qualities that are not inherent in the elements that make up the system. The term “system effect” is also used as a synonym for emergence. The history of the origin and development of the concept of emergence within the framework of general systems theory, as well as the basic ideas underlying this phenomenon, are presented in [1]. In particular, with reference to [2], it is indicated that emergent behavior can be understood based on the nature and behavior of its parts plus knowledge of how these parts interact.

A large number of works are devoted to the philosophical and general methodological understanding of this phenomenon as a whole [3]. In [4], the categories of strong and weak emergence are substantiated; these concepts are quite conditional, and it is difficult to draw a clear boundary between them. Moreover, it is believed that emergence can manifest itself on a continuum from “weak” to “strong” [5]. Emergence is considered strong if there is no acceptable theory that could explain or derive the behavior of the system based on the properties or behavior of its components. On the contrary, emergence is weak if the dependence between the behavior of the components and the behavior of the system is clearly observed. In [6], an attempt is made to mathematically prove the possibility of defining the concept of strong emergence. The paper [7] is devoted to the formalization of the concept of emergence based on entropy. It is argued that there is a connection between entropy and emergence in complex systems in the sense that an increase in entropy is an indicator of the possible occurrence of emergent events and states. In conclusion, the authors formulate the question of how to express the entropy of a system through the entropy of its constituent subsystems.

Attempts to give a formal description of such effects in specific physical or engineering systems cause significant difficulties, so there are far fewer examples of such studies. Here it is worth noting the work [8], which examines the occurrence of emergent effects in artificial systems, as well as the work [9], devoted to the study of principles and approaches to the construction of engineering systems with such properties. Specific examples of the manifestation of the system effect in real physical systems can be found, for example, in [10–13].

Due to the lack of a clear understanding, emergence is defined differently in different studies. For example, in [14, 15] it is understood as the ability of interacting autonomous agents to form coordinated behavior without external control, but only on the basis of internal interaction. A similar view is reflected in [16], which examines the processes of selforganization and emergence in the formation of a common vocabulary at the primary stages of the formation of human communities. The main idea is also based on modeling the interaction of agents acting on the basis of a small number of simple rules, which leads to emergent effects.

The study of emergence is closely related to the study of processes that lead to complex behavior of a system with relatively simple behavior of its elements. Perhaps the formal justification of such processes even in relatively narrow specific cases of such systems is the key issue in understanding the nature of emergence as such. These issues are central to the publication [17].

The system effect is found in systems of various natures – polymetallic compounds, chemical processes, collective decision-making in social structures, etc. The diversity of its manifestations predetermines the formal methods that are most suitable for research in each case. These may be equations describing the elastic interaction of solids, laws of chemical reactions, or algorithms for making compromise decisions, in particular, algorithms for group control in multiagent systems. The study of system effects involves the preliminary construction of a formal model that should correctly predict the behavior of the object or process under permissible impacts on it.

It is generally accepted that for the existence of any system, system-forming factors are necessary, which contribute to the formation of the system, are external in relation to its elements, and are not caused by the need for unification [18]. In many cases, the role of such factors is played by connections between elements, the nature of which is determined by the nature of the elements that form the system.

The basic assumption that motivated this work is that a fairly universal way to study emergence is to study the mappings realized by a complex system depending on the mappings realized by the elements of the system and on the state of the system-forming factors, i.e., the structure of the connections between the elements of the system. This is almost obvious in the case of information processing systems, including those that include people.

Below we consider a class of systems implementing binary mappings. In a certain sense, they can be considered as automaton mappings. Despite their apparent simplicity, automaton mappings are in many cases not inferior to classical algebraic structures in terms of their breadth of application and expressiveness. It is also known that semigroups, groups and rings of automata, as well as their functional systems, have contributed to the solution of a number of difficult abstract mathematical problems.

Models of binary mapping systems do not provide a complete explanation of all processes of system emergence. However, they demonstrate important properties that are inherent in this phenomenon. For example, the so-called “superadditive” effect that occurs when combining systems, as well as the emergence of limit cycles in a set of states. In physical systems, limit cycles correspond to attractors that can be considered as possible macrostates in multiscale systems. In the case of binary mapping systems, it becomes possible to move to strictly formal models and obtain a number of quantitative characteristics of processes that reflect emergence. It should be noted that superadditive effects on classes of binary mappings, according to the almost generally accepted

terminology and classification, can be attributed to the class of weak emergence [4–6], since there is a high level of traceability between the functions of the elements and the functions of the system as a whole.

The paper is structured as follows. Section 2 provides the basic definitions, notations, and terms that are used further in the text. Sections 3 and 4 contain the derivation of the basic relations describing the realized binary mappings in systems with the structure of a complete graph and an arbitrary directed graph, respectively. Section 5 proves statements describing the classes of binary mappings that are obtained as a result of set-theoretic operations on systems. The resulting system effect is also described. Section 6 is devoted to the definition and justification of the emergence coefficient for binary systems, as well as the proof of two-sided estimates for it.

2. BASIC DEFINITIONS AND NOTATIONS

We will use the *Kronecker product* $A \otimes B$ of matrices A and B , namely, if $A = (a_{ij})$ is a matrix of dimensions $m \times n$, and B is a matrix of dimensions $p \times q$, then by definition the matrix $A \otimes B = (a_{ij}B)$ and obviously has dimensions $mp \times nq$. The definition and properties of the Kronecker product can be found, for example, in [19]. In particular, we will need the equality $(A \otimes B)(C \otimes D) = AC \otimes BD$ connecting the ordinary and Kronecker products, where matrices A , B , C and D have sizes for which all products are defined. This property can be generalized to an arbitrary number of factors. For convenience, we will denote the Kronecker product of matrices $A_1, A_2, \dots, A_k$ by $A_1 \otimes A_2 \otimes \dots \otimes A_k = \bigotimes_{i=1}^k A_i$. In particular, we will write the Kronecker k th power of a matrix A as $\underbrace{A \otimes \dots \otimes A}_k = A^{[k]}$. It is important to take into account the non-commutativity of the Kronecker product.

The *Kronecker extension* of a vector $a = (a_1 \ a_2 \ \dots \ a_n)^\top$ is the vector

$$\bar{a} = (1 \ a_1 \ a_2 \ (a_1 a_2) \ \dots \ (a_1 a_2 \ \dots \ a_n))^\top = \begin{pmatrix} 1 \\ a_n \end{pmatrix} \otimes \begin{pmatrix} 1 \\ a_{n-1} \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 1 \\ a_1 \end{pmatrix}.$$

The *Kronecker extension* of a matrix A by rows is the matrix $\bar{A}$, each row of which, considered as a vector, is the Kronecker extension of the corresponding row of the matrix A . The *Hadamard product* of matrices A and B is the matrix $A * B$, obtained as a result of element-wise multiplication of these matrices [19]. This product is defined for matrices of the same size and is a commutative operation. The matrix is called *monomial by columns* if each of its columns consists of zeros except for one element, which is equal to one.

Below we will need some basic definitions from graph theory. Let $H = (V, E)$ be a directed graph, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and $E = \{e_1, e_2, \dots, e_p\} \subseteq V \times V$ is the set of arcs. The graph is allowed to contain loops, i.e., arcs of the form $e = (v, v)$. If $v_i \in V$, then ρ_i will denote the *indegree* of the vertex v_i , i.e., the number of arcs of the form (v, v_i) entering this vertex, so that $\sum_{i=1}^n \rho_i = |E|$. A graph with n vertices that contains all possible arcs and loops, i.e., when $E = V \times V$, $|E| = n^2$, is called a *complete directed graph with loops* (or simply a *complete graph* for short), and will be denoted by K_n . If $E = \emptyset$, then such a graph is called a *null graph* and is denoted by O_n .

The *adjacency matrix* of a directed graph H is a square matrix $R = (r_{ij})$ of size $n \times n$, in which $r_{ij} = 1$, if $(v_i, v_j) \in E$, and $r_{ij} = 0$ otherwise. The *union* and *intersection* of graphs $H_1 = (V, E_1)$ and $H_2 = (V, E_2)$ with the same set of vertices are graphs $H_1 \cup H_2 = (V, E_1 \cup E_2)$ and $H_1 \cap H_2 = (V, E_1 \cap E_2)$, respectively.

3. SYSTEMS WITH COMPLETE GRAPH STRUCTURE

By a *system* we mean a set of n interacting functional elements that operate in discrete time $t = 0, 1, 2, \dots$ and can be in one of two *states* – 0 or 1. The state z_i of the i th element at a given time $t + 1$ is determined by a Boolean function $z_i(t + 1) = f_i(x_1(t + 1), \dots, x_m(t + 1); z_1(t), \dots, z_n(t))$ that depends on two groups of arguments. The arguments of the first group are external arguments $x_1(t + 1), \dots, x_m(t + 1)$ that are binary external signals, the arguments of the second group are internal arguments $z_1(t), \dots, z_n(t)$ that are the states of the corresponding elements of the system at a given time t . We will call the functions z_i , $i = 1, \dots, n$, the *state functions* of the elements, and the ordered set of states of all elements — the *state of the system*. We assume that the system does not depend on random factors.

The sequence of states of the system can be considered as a binary mapping of the sequence of external arguments and previous states. In general, the functions implemented by elements may not depend on the states of some other elements, which is determined by the structure of connections between elements. Therefore, the sequence of states depends on: 1) the functions implemented by each element of the system, and 2) the structure of connections between elements. If the system contains n elements with m inputs, then each element can be described by one of the 2^{m+n} Boolean functions.

The structure of the system is conveniently defined using a directed graph, the vertices of which are in one-to-one correspondence with the elements of the system, and the arcs correspond to the connections between the outputs of some elements and the inputs of others. Since the state of each element can also depend on its own state, the system graph can have loops. The described system can be considered as a synchronous *sequential* logical network implementing an *automaton* mapping [20].

Let us consider a system of n elements with structure K_n . The mapping realized by it can be described by a system of state functions, in which the right-hand sides are some Boolean functions:

$$z_i(t + 1) = f_i(x_1(t + 1), \dots, x_m(t + 1), z_1(t), \dots, z_n(t)), \quad i = 1, \dots, n.$$

Next, we will consider the case when there are no external arguments, i.e., $z_i(t + 1) = f_i(z_1(t), \dots, z_n(t))$, $i = 1, \dots, n$. By analogy with differential equations, such a system can be called autonomous, describing its “own free movement” or, in the terminology of systems analysis, a closed system demonstrating its behavior depending on the initial state $z_1(0), \dots, z_n(0)$. We will describe the mappings realized by such a system for all possible state functions of the system’s elements.

As is known [21–23], any Boolean function of n variables can be uniquely defined by its *arithmetic representation*:

$$z_i(t + 1) = f_i(z_1(t), \dots, z_n(t)) = a_0^i + a_1^i z_1(t) + a_2^i z_2(t) + a_{12}^i z_1(t) z_2(t) + \dots + a_{k_1 \dots k_p}^i z_{k_1}(t) \cdot \dots \cdot z_{k_p}(t) + \dots + a_{1 \dots n}^i z_1(t) \cdot \dots \cdot z_n(t), \quad i = 1, \dots, n, \quad (1)$$

where $a_0^i, a_1^i, a_2^i, a_{12}^i, \dots, a_{k_1 \dots k_p}^i, \dots, a_{1 \dots n}^i$ are integers that uniquely define the function f_i .

Let us consider the Kronecker extension of the vector $z = (z_1 \ z_2 \ \dots \ z_n)^\top$:

$$\bar{z} = (1 \ z_1 \ z_2 \ (z_1 z_2) \ \dots \ (z_1 z_2 \dots z_n))^\top = \begin{pmatrix} 1 \\ z_n \end{pmatrix} \otimes \begin{pmatrix} 1 \\ z_{n-1} \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 1 \\ z_1 \end{pmatrix}.$$

Then the system of relations (1) can be written in matrix form:

$$z(t + 1) = A\bar{z}(t), \quad (2)$$

here A is the matrix of the size $n \times 2^n$ of the corresponding coefficients.

Equality (2) can be written differently. The vector $\bar{z}(t)$ takes 2^n different values, and each of them is assigned an image under the mapping A . Therefore, we can write 2^n equalities (2) corresponding to different $\bar{z}(t)$. It is easy to prove by induction that all possible 2^n values of the vector $\bar{z}(t)$, written in the natural order, form a matrix $Q_n = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{[n]}$. Let us denote by B the matrix whose columns are the images of the corresponding vectors $\bar{z}(t)$. Obviously, this matrix completely and uniquely describes the binary mapping realized by the system. Then all possible 2^n equalities (2) can be written compactly as $B = AQ_n$. Since $Q_n^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}^{[n]}$, we immediately obtain

$$A = BQ_n^{-1}. \quad (3)$$

This formula gives an explicit expression for the coefficients of the arithmetic representation of the state functions of all elements of the system that realizes the binary mapping specified by the matrix B . The uniqueness of solution (3) means that any mapping can be realized in a system with structure K_n . In other words, the *class of mappings* realized in such a system coincides with the set of all possible mappings. If the structure of the system differs from K_n , then this will no longer be true.

Next, we describe the classes of realizable binary mappings depending on the structure of the system, defined by an arbitrary directed graph (not necessarily a graph K_n).

4. SYSTEMS WITH ARBITRARY STRUCTURE AND CLASSES OF MAPPINGS

Let us consider the general case when the system has an arbitrary structure defined by a directed graph $H = (V, E)$. The state function of the i th element depends on the state of the j th element if two conditions are simultaneously met: 1) there is a connection between the input of the i th element and output of the j th element and 2) the value of the state of the j th element is an essential variable for the i th element. The first condition is determined by the structure of the system, and the second by the state function. This reflects the relationship between the intra-element and supra-element organization of the system. A system with a structure K_n is maximally “informed,” since each element has information about the states of all other elements (including itself) and can be configured to perform any function.

Let us characterize the class of mappings realized by a system with an arbitrary structure H . Let $R = (r_{ij})$ be the adjacency matrix of the graph H . Using the arithmetic representation (1), we obtain a matrix relation $z(t+1) = A\bar{z}(t)$ similar to (2), where, as before, A is a matrix, each row of which is formed by the coefficients of the corresponding arithmetic representations.

However, in the general case $z_i(t+1)$ does not depend on all values of $z_j(t)$, $j = 1, \dots, n$, but only on some of them, determined by the structure of the system. More precisely, $z_i(t+1)$ essentially depends on $z_j(t)$ if $r_{ji} = 1$, and, accordingly, does not depend if $r_{ji} = 0$. Thus, all information about the structural dependence of the i th element on the other elements is contained in the i th column of the matrix R . If $z_i(t+1)$ does not structurally depend on $z_j(t)$, then we can assume that the coefficient a_j^i in the expansion is equal to zero or, in the general case, we can write $a_j^i = r_{ji}u_j^i$. It is clear that in the remaining terms of the expansion, for example, $a_{k_1 \dots k_2}^i z_{k_1}(t) \cdot \dots \cdot z_{k_2}(t)$, the coefficient $a_{k_1 \dots k_2}^i$ is also equal to zero if at least one of the connections of the k_1 th, $\dots$, k_p th elements with the input of the i th element is absent. This is equivalent to the equality: $a_{k_1 \dots k_p}^i = r_{k_1 i} \cdot \dots \cdot r_{k_p i} u_{k_1 \dots k_p}^i$. Consequently, the matrix A can be written as a Hadamard product

of two matrices:

$$A = \begin{pmatrix} a_0^1 & a_1^1 & a_2^1 & a_{12}^1 & \dots & a_{1\dots n}^1 \\ a_0^2 & a_1^2 & a_2^2 & a_{12}^2 & \dots & a_{1\dots n}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_0^n & a_1^n & a_2^n & a_{12}^n & \dots & a_{1\dots n}^n \end{pmatrix} \\ = \begin{pmatrix} u_0^1 & u_1^1 & u_2^1 & u_{12}^1 & \dots & u_{1\dots n}^1 \\ u_0^2 & u_1^2 & u_2^2 & u_{12}^2 & \dots & u_{1\dots n}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u_0^n & u_1^n & u_2^n & u_{12}^n & \dots & u_{1\dots n}^n \end{pmatrix} * \begin{pmatrix} 1 & r_{11} & r_{21} & (r_{11} \cdot r_{21}) & \dots & (r_{11} \cdot \dots \cdot r_{n1}) \\ 1 & r_{12} & r_{22} & (r_{12} \cdot r_{22}) & \dots & (r_{12} \cdot \dots \cdot r_{n2}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & r_{1n} & r_{2n} & (r_{1n} \cdot r_{2n}) & \dots & (r_{1n} \cdot \dots \cdot r_{nn}) \end{pmatrix}.$$

The first matrix in the product, denoted by U , is actually the matrix of the coefficients of the representation. The second matrix reflects the structural dependence of the elements of the system. It is obviously the Kronecker expansion over the rows $\overline{R^\top}$ of the transposed adjacency matrix R . The matrix $\overline{R^\top}$, obviously consists of zeros and ones, as does the matrix R . In view of this, the matrix relation can be written as follows:

$$z(t+1) = (U * \overline{R^\top}) \bar{z}(t). \quad (4)$$

This expression defines the mappings realized by a system with an arbitrary structure H , determined by the adjacency matrix R . In particular, if $H = K_n$, all elements of the matrix R , and therefore of the matrix $\overline{R^\top}$, are equal to one, and formula (4) coincides with formula (2).

Just as expression (3) was obtained from (2), for the coefficients of the arithmetic representation that determines the matrix A , from relation (4) we can express $U * \overline{R^\top}$:

$$U * \overline{R^\top} = BQ_n^{-1}. \quad (5)$$

However, unlike the previous case, in this relation the matrix B cannot be arbitrary; it must be a binary matrix, but such that the elements on the right-hand side of (5) are guaranteed to be equal to zero, the locations of which are determined by the position of the zero elements of the matrix $\overline{R^\top}$.

Each state of the system can be identified with a binary vector. If we write all possible states of the system in the natural order, they form a binary matrix G of size $n \times 2^n$. For $n = 2, 3$ these matrices are shown below:

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Each column of the matrix B is one of the possible states of the system, so we can write $B = GP$, where P is a column-monomial matrix of size $2^n \times 2^n$. In other words, GP is the same matrix B , as in the expression for the coefficients of the arithmetic representation (3), but written explicitly using the monomial matrix P , i.e., the set of matrices B and all monomial matrices P are in one-to-one correspondence. This allows us to further operate not with all possible images of binary vectors in the mapping, but with all column-monomial matrices, which is much more convenient from a formal point of view. Substituting this equality into formula (5) for the matrix of coefficients of the arithmetic representation in the case of an arbitrary structure, we obtain

$$U * \overline{R^\top} = GPQ_n^{-1}. \quad (6)$$

Further, we will call this equality the *relation for the coefficients*. If the matrix P satisfies this condition for some matrix U , then it completely describes the binary mapping realized by the system with the given structure H . Thus, there is a one-to-one correspondence between mappings and monomial matrices satisfying the relation for the coefficients (6) for some matrix U . Therefore, each mapping can be identified with the corresponding monomial matrix P .

Let us denote by $\mathfrak{P}$ the set of all monomial matrices of size $2^n \times 2^n$. Obviously, $\mathfrak{P}$ is a semigroup with unity, which contains, in particular, a subsemigroup isomorphic to the symmetric group S_{2^n} . The set $\mathcal{P}(H) \subseteq \mathfrak{P}$ of transformations P , satisfying the relation for the coefficients (6) for some matrix U , determines the *class of mappings* realized by a system with the graph structure H . This can be explained as follows.

The mapping of all possible states of the system is completely determined by the set of specific mappings realized by each element of this system, i.e., by the Boolean functions describing their functioning. In turn, these functions are completely determined by the corresponding coefficients of their arithmetic representations. But these coefficients are uniquely determined by the matrix B , and, due to what has been said above, by the corresponding monomial matrix P . In the case of a complete graph structure, for any possible mapping of the system states, it is possible to calculate the coefficients of the arithmetic representations of the functions of the elements, under which the system will realize this mapping.

For a system with an arbitrary structure this is not so, since the functions realized by the elements of the system, and therefore the system as a whole, are limited by the existing structure of the system, i.e., the connections between the elements. Therefore, not every mapping can be realized in such a system. Since the mapping of the states of the system is completely determined by the monomial matrix P , this means that the functions of the elements, and therefore the coefficients of their arithmetic representations, cannot be found for every matrix so that the relation for the coefficients (or equality) (6) is satisfied. All possible mappings that can be realized in a system with a fixed structure due to all possible choices of the functions of the elements form a class of mappings for a given structure.

Let us characterize the matrices P included in the set $\mathcal{P}(H)$. For this, we will need an auxiliary “index” set. For convenience, let the matrices $\overline{R}^\top$ and U be specified using the usual two-index numbering, i.e., $\overline{R}^\top = (\bar{r}_{ij})$, $U = (u_{ij})$, $i = 1, \dots, n$, $j = 1, \dots, 2^n$. Let us introduce the set $\text{Ind}\overline{R}^\top = \{(i, j) \mid i = 1, \dots, n, j = 1, \dots, 2^n, \bar{r}_{ij} = 0\}$. In other words, $\text{Ind}\overline{R}^\top$ is formed by all possible pairs of indices that determine the positions of the zero elements of the matrix $\overline{R}^\top$. The following statement gives a description of the set $\mathcal{P}(H)$, that does not depend on U .

Statement 1. $\mathcal{P}(H) = \{P \mid P \in \mathfrak{P}, g_i P q_j^{(-1)} = 0, (i, j) \in \text{Ind}\overline{R}^\top\}$.

Proof. Let g_i and $q_j^{(-1)}$ denote the i th row in G and the j th column in Q_n^{-1} . Then the (i, j) -element of the matrix in the relation for the coefficients (6) on the right is obviously equal to $g_i P q_j^{(-1)}$. If the corresponding element of the matrix $\overline{R}^\top$ is equal to zero, i.e., $\bar{r}_{ij} = 0$, then the (i, j) -element of the matrix $U * \overline{R}^\top$ is also equal to zero. Therefore, in order for the matrix equality for the coefficients (6) to hold, it must be $g_i P q_j^{(-1)} = 0$. Otherwise, i.e., for $\bar{r}_{ij} = 1$, the (i, j) -element of the matrix $U * \overline{R}^\top$ is equal to the (i, j) -element of the matrix U . Since this element is one of the coefficients, then, setting it equal to $g_i P q_j^{(-1)}$, we obtain the required equality of the elements on the left and right. Therefore, a monomial matrix $P \in \mathfrak{P}$ satisfies the relation for the coefficients (6) if it satisfies the system of relations $g_i P q_j^{(-1)} = 0$, $(i, j) \in \text{Ind}\overline{R}^\top$, which proves the statement.

Each structure H generates a class $\mathcal{P}(H)$ of binary mappings formed under all possible functional settings of the system elements. In particular, if $H = K_n$, then R is a matrix of ones. In this case, obviously, $\mathcal{P}(K_n) = \mathfrak{P}$, and hence any mapping can be realized in such a system. If the structure of the system is a null graph, i.e., $H = O_n$, then $R = \mathbf{0}$, and $\mathcal{P}(O_n)$ contains 2^n matrices P , characterized by the fact that the elements of one of the rows are equal to one, and the rest are equal to zero.

The following statement gives an explicit expression for the cardinality $|\mathcal{P}(H)|$ of the class of mappings $\mathcal{P}(H)$, realized by a system with structure H .

Statement 2. *Let the structure of the system be defined by a directed graph $H = (V, E)$, $|V| = n$, ρ_i is the indegree of a vertex i in the graph H , $i = 1, \dots, n$. Then*

$$|\mathcal{P}(H)| = 2^{\sum_{i=1}^n 2^{\rho_i}}. \quad (7)$$

Proof. The number of different binary mappings realized by the i th element with ρ_i inputs for all possible functional settings is $2^{2^{\rho_i}}$. Therefore $|\mathcal{P}(H)| = 2^{2^{\rho_1}} \cdot 2^{2^{\rho_2}} \cdot \dots \cdot 2^{2^{\rho_n}} = 2^{\sum_{i=1}^n 2^{\rho_i}}$.

Corollary 1. 1) $|\mathcal{P}(K_n)| = 2^{n2^n}$; 2) $|\mathcal{P}(O_n)| = 2^n$.

The next section describes the classes of binary mappings that result from set-theoretic operations on systems.

5. OPERATIONS ON SYSTEMS AND CLASSES OF REALISABLE MAPPINGS

Let there be two systems with structures H_1 and H_2 . The question arises: are there classes of mappings that, given suitable state functions of the elements, can be realized in both the first and second systems? If so, how can they be characterized? Let us examine this question.

Let us recall once again that, as described above, each mapping of possible states of a system with a given structure and fixed functions of the elements is uniquely described by a column-monomial matrix P . For all possible functions of the elements, a class of mappings is obtained that are realized by a system with a given structure. Therefore, the description of classes of mappings is reduced to the description of all possible and admissible monomial matrices.

If some mapping defined by the matrix P , is realized by both the first and the second systems, then $P \in \mathcal{P}(H_1)$ and $P \in \mathcal{P}(H_2)$, and therefore $P \in \mathcal{P}(H_1) \cap \mathcal{P}(H_2)$. The converse is also true: any matrix P , lying in the intersection of the sets $\mathcal{P}(H_1)$ and $\mathcal{P}(H_2)$, determines a mapping realized by each system.

Statement 3. $\mathcal{P}(H_1 \cap H_2) = \mathcal{P}(H_1) \cap \mathcal{P}(H_2)$ for any H_1 and H_2 .

Proof. Let $R_1 = (r_{ij}^{(1)})$ and $R_2 = (r_{ij}^{(2)})$ be the adjacency matrices of graphs H_1 and H_2 . Then $\mathcal{P}(H_1)$ is determined by the system of relations $g_i P q_j^{(-1)} = 0$, $(i, j) \in \text{Ind} \overline{R_1^\top}$, and $\mathcal{P}(H_2)$ is determined by the system of relations $g_i P q_j^{(-1)} = 0$, $(i, j) \in \text{Ind} \overline{R_2^\top}$. If $P \in \mathcal{P}(H_1) \cap \mathcal{P}(H_2)$, then this means that P must satisfy each of these systems or, in other words, must satisfy the system of relations $g_i P q_j^{(-1)} = 0$, $(i, j) \in \text{Ind} \overline{R_1^\top} \cup \text{Ind} \overline{R_2^\top}$.

Let us introduce the matrix $\overline{R^\top}$, which is defined as follows: (i, j) -element of this matrix is zero if the corresponding element of at least one of the matrices $\overline{R_1^\top}$ or $\overline{R_2^\top}$, is zero, and one when both elements are one. Then, obviously, $\text{Ind} \overline{R^\top} = \text{Ind} \overline{R_1^\top} \cup \text{Ind} \overline{R_2^\top}$. It follows directly from the definition of the matrix $\overline{R^\top}$ that $\overline{R^\top} = \overline{R_1^\top} * \overline{R_2^\top}$. Let us now show that $\overline{R^\top}$ is a Kronecker extension along the rows of the matrix $R^\top = (R_1 * R_2)^\top = R_1^\top * R_2^\top$, i.e., $\overline{R^\top} = \overline{R_1^\top} * \overline{R_2^\top}$. To do this, it suffices to prove the relation

$$\overline{R_1^\top * R_2^\top} = \overline{R_1^\top} * \overline{R_2^\top}. \quad (8)$$

First of all, note that the first columns of these matrices are equal, since they contain only ones. Let us show that the remaining elements are also equal. An arbitrary element of the matrix on the left has the form $(r_{k_1 i}^{(1)} r_{k_1 i}^{(2)}) \cdot \dots \cdot (r_{k_p i}^{(1)} r_{k_p i}^{(2)})$, $p = 1, \dots, n$, and the corresponding element of the right matrix is $(r_{k_1 i}^{(1)} \cdot \dots \cdot r_{k_p i}^{(1)}) \cdot (r_{k_1 i}^{(2)} \cdot \dots \cdot r_{k_p i}^{(2)})$. Both expressions differ only in the order of the

factors, and therefore are equal. Therefore, equality (8) is true. But $R = R_1 * R_2$ is the adjacency matrix of the graph $H = H_1 \cap H_2$, from which the validity of the statement follows.

Corollary 2. $\mathcal{P}(H_1 \cap \dots \cap H_k) = \mathcal{P}(H_1) \cap \dots \cap \mathcal{P}(H_k)$ for any $H_1, \dots, H_k$.

This corollary is proved by induction. Statement 3 establishes a precise connection between classes of binary mappings and systems at their intersection. A similar relation for the union of systems is generally incorrect, but the following inclusion holds.

Statement 4. $\mathcal{P}(H_1 \cup H_2) \supseteq \mathcal{P}(H_1) \cup \mathcal{P}(H_2)$ for any H_1 and H_2 .

Proof. Using Statement 3, we have

$$\mathcal{P}(H_1 \cup H_2) \cap \mathcal{P}(H_1) = \mathcal{P}((H_1 \cup H_2) \cap H_1) = \mathcal{P}(H_1).$$

It follows that $\mathcal{P}(H_1) \subseteq \mathcal{P}(H_1 \cup H_2)$. Similarly, we obtain $\mathcal{P}(H_2) \subseteq \mathcal{P}(H_1 \cup H_2)$. The validity of the statement follows from these two inclusions.

The substantive meaning of Statement 4 is that the set of mapping classes realized by a system with structure $H = H_1 \cup H_2$, is greater than the simple union of mapping classes of the systems being combined. This fact, in a certain sense, reflects the “superadditive effect” inherent in systems, corresponding to the thesis “the whole is greater than the sum of its parts.” The mapping classes realized in a system with a graph structure H and not realized in any of the systems being combined demonstrate the effect of emergence, understood in the narrow sense. This statement can be easily generalized to any number of systems.

Corollary 3. $\mathcal{P}(H_1 \cup \dots \cup H_k) \supseteq \mathcal{P}(H_1) \cup \dots \cup \mathcal{P}(H_k)$ for any $H_1, \dots, H_k$.

From Statement 4 follows the property of monotonicity with respect to inclusion of systems.

Corollary 4. It follows from $H_1 \subseteq H_2$ that $\mathcal{P}(H_1) \subseteq \mathcal{P}(H_2)$.

Let us characterize in more detail the left and right parts of Statement 4.

Statement 5. The set $\mathcal{P}(H_1 \cup H_2)$ consists of those and only those matrices $P \in \mathfrak{P}$, that satisfy the system of relations $g_i P q_j^{(-1)} = 0$, $(i, j) \in \text{Ind}(\overline{R_1 + R_2 - R_1 * R_2})^\top$.

Proof. Let $H = H_1 \cup H_2$. It is obvious that the adjacency matrix R of the structure H is connected with the adjacency matrices R_1 and R_2 of the structures H_1 and H_2 by the equality $R = R_1 + R_2 - R_1 * R_2$. In view of this, the validity of the statement being proved follows from Statement 1.

Statement 6. The set $\mathcal{P}(H_1) \cup \mathcal{P}(H_2)$ consists of those and only those matrices $P \in \mathfrak{P}$, that satisfy the system of relations

$$g_i P q_j^{(-1)} g_k P q_l^{(-1)} = 0, \quad (i, j) \in \text{Ind} \overline{R_1}^\top, \quad (k, l) \in \text{Ind} \overline{R_2}^\top. \quad (9)$$

Proof. If $P \in \mathcal{P}(H_1)$ or $P \in \mathcal{P}(H_2)$, then P satisfies the system of relations (9). Conversely, if P does not belong to, for example, $\mathcal{P}(H_1)$, but satisfies (9), then for some $(i, j) \in \text{Ind} \overline{R_1}^\top$ we have $g_i P q_j^{(-1)} \neq 0$. If we consider the equations of the system of relations (9), in which the indices i and j coincide with the chosen ones, i.e., $g_i P q_j^{(-1)} g_k P q_l^{(-1)} = 0$, $(k, l) \in \text{Ind} \overline{R_2}^\top$, then we obtain the relations $g_k P q_l^{(-1)} = 0$, $(k, l) \in \text{Ind} \overline{R_2}^\top$. Consequently, $P \in \mathcal{P}(H_2)$. Similarly, the matrix P , which does not belong to $\mathcal{P}(H_2)$ and satisfies (9), belongs to $\mathcal{P}(H_1)$, which proves the statement.

6. EMERGENCE COEFFICIENT

As shown above, when combining systems, a “superadditive” property appears, which is expressed in the fact that the new system is capable of realizing mappings that are not realizable by

any of the original systems. In order to quantitatively characterize this effect, it is necessary to introduce a suitable scalar indicator that can be interpreted as the degree of irreducibility of the properties of the system to the properties of its elements.

The simplest and most straightforward approach is to use the ratio $\frac{|\mathcal{P}(H_1 \cup H_2)|}{|\mathcal{P}(H_1) \cup \mathcal{P}(H_2)|}$, which shows the multiplicity of the excess of the number of implemented mappings when combining systems. However, despite its simplicity, such an indicator is difficult to interpret. This is due to the fact that it takes very large values due to the twice exponential growth of the cardinality of the classes of these mappings, as established in Statement 2. The following considerations suggest a slightly different approach to constructing such an indicator. Let us consider simple examples.

Let the graph of the structure of a three-element system have $n = 3$ vertices and $\rho_1 = \rho_2 = 1$, $\rho_3 = 2$ be the indegrees of its vertices. Then, by virtue of Statement 2, the cardinality of the class of realizable mappings is equal to $2^{2^1+2^1+2^2} = 2^8 = 256$. On the other hand, the class of mappings realizable by a system with a structure K_2 with two elements will have exactly this cardinality, i.e., $2^{2 \cdot 2^2} = 2^8 = 256$.

Another similar example. The graph H has $n = 6$ vertices with indegrees 2, 1, 1, 2, 3, 2. Then $|\mathcal{P}(H)| = 2^{2^2+2^1+2^1+2^2+2^3+2^2} = 2^{24}$. But the class of mappings of the same power is realized by the structure K_3 , i.e., $|\mathcal{P}(K_3)| = 2^{3 \cdot 2^3} = 2^{24} = |\mathcal{P}(H)|$. In other words, in each of these cases for some μ the equality $2^{\mu 2^\mu} = 2^{2^{\rho_1} + 2^{\rho_2} + \dots + 2^{\rho_n}}$ is satisfied or

$$\mu 2^\mu = 2^{\rho_1} + 2^{\rho_2} + \dots + 2^{\rho_n}. \quad (10)$$

In these examples, μ is the smallest number of vertices of a complete graph with loops that realizes exactly the same number of mappings as some graph with given indegrees of vertices. In the cases considered, μ was an integer and could be interpreted as the number of vertices of a complete directed graph. In the general case, the solution μ of equation (10) for arbitrary $\rho_1, \rho_2, \dots, \rho_n$ will not be an integer. Nevertheless, speculatively reasoning, we can assume that the solution of equation (10) is a scalar characteristic of the considered class of mappings as a minimal, but “fuzzy” (not integer) number of vertices of a complete directed graph with loops that realizes the same number of different mappings. Since the function $\mu 2^\mu$ is positive and monotonically increasing on $[0, \infty)$, then for any right-hand side, equation (10) has a unique root.

The quantity μ can be considered as a measure $\mu = \mu[\mathcal{P}(H)]$ of a set $\mathcal{P}(H)$ on the Boolean $2^\mathfrak{P}$, where $\mathfrak{P}$, as defined above, is the set of all monomial matrices of the corresponding size. This measure is obviously not additive, but is *monotone* with respect to inclusion of classes, since $\mu[\mathcal{P}(H_1)] \leq \mu[\mathcal{P}(H_2)]$ for $\mathcal{P}(H_1) \subseteq \mathcal{P}(H_2)$.

By the *emergence coefficient* $\kappa(H_1, H_2)$ when uniting systems H_1 and H_2 we will call the quantity $\kappa(H_1, H_2) = \frac{\mu[\mathcal{P}(H_1 \cup H_2)]}{\mu[\mathcal{P}(H_1) \cup \mathcal{P}(H_2)]}$. From this definition it follows that $\mu[\mathcal{P}(H_1 \cup H_2)] = \mu_1$ and $\mu[\mathcal{P}(H_1) \cup \mathcal{P}(H_2)] = \mu_2$ are the roots of the corresponding equations

$$\mu_1 2^{\mu_1} = \log_2 |\mathcal{P}(H_1 \cup H_2)|, \quad (11)$$

$$\mu_2 2^{\mu_2} = \log_2 |\mathcal{P}(H_1) \cup \mathcal{P}(H_2)| = \log_2 (|\mathcal{P}(H_1)| + |\mathcal{P}(H_2)| - |\mathcal{P}(H_1 \cap H_2)|). \quad (12)$$

The introduced concept of the emergence coefficient can easily be generalized to an arbitrary number of uniting systems. The following theorem gives two-sided estimates of the introduced coefficient.

Theorem 1. *The emergence coefficient $\kappa(H_1, H_2)$ for the union of any two n -element systems H_1 and H_2 satisfies the inequalities*

$$1 \leq \kappa(H_1, H_2) < 1 + \sqrt{1 + \frac{1}{e \ln 2}} \approx 2.2372.$$

The obtained upper bound is quite accurate, which is especially noticeable as n increases. The table below shows the values of the maximum emergence coefficient for $n = 2, \dots, 20$ (accurate to four decimal places). Comparison of the given data with the found upper bound allows us to make an assumption about its asymptotic accuracy.

Table 1. Values of the maximum emergence coefficient

n	Maximum emergence coefficient	n	Maximum emergence coefficient
		11	2.0822
2	1.2553	12	2.1245
3	1.3729	13	2.1567
4	1.4979	14	2.1807
5	1.5981	15	2.1980
6	1.6946	16	2.2102
7	1.7869	17	2.2184
8	1.8803	18	2.2236
9	1.9610	19	2.2265
10	2.0281	20	2.2278

7. CONCLUSION

The paper proposes an apparatus for studying system effects arising in arbitrary binary mapping systems. Despite the binary nature of the implemented mappings, the standard algebraic technique is used for the study, which simplifies the description and understanding of emergence effects in the narrow sense. The analysis allows us to conclude that the classes of mappings observed in such systems and their transformations during set-theoretical operations on the structures of these systems are similar to the “expansion effect” when combining classical algebraic structures. The results obtained, in particular, Statements 3, 4 and their consequences, as well as estimates for the emergence coefficient given by Theorem 1, allow us to establish in which cases and what quantitatively the emergent effect can be achieved when aggregating binary systems. This can be used, for example, to optimize the collective behavior of individual groups of functioning objects or subjects.

The model of binary mappings considered in the work is a fairly narrow class of mappings. However, due to its relative simplicity, it allows us to identify some patterns inherent in system interactions and even characterize them quantitatively. This, in fact, was the motivation for this work. Generalization of these studies to a more general class of system interactions (even in the autonomous case) will require the use of a more complex mathematical apparatus associated with morphisms on differential manifolds and the construction of suitable measures on such objects. This is the topic of a separate complex study. But such a theory can provide a more natural description of emergence effects in physical environments and in technical systems.

APPENDIX

Proof of Theorem 1. The left inequality follows immediately from Statement 4 and the monotonicity of the function $x2^x$. Let us prove the right inequality.

Let $H_1 = (V, E_1)$ and $H_2 = (V, E_2)$ be arbitrary directed graphs and $|V| = n$. Consider the sets of arcs $E_C = E_1 \cap E_2$, $E_A = E_1 \setminus E_C$ and $E_B = E_2 \setminus E_C$. These sets are obviously pairwise disjoint. Let us construct three graphs $H_A = (V, E_A)$, $H_B = (V, E_B)$ and $H_C = (V, E_C)$; let $\rho_i^A, \rho_i^B, \rho_i^C, i = 1, \dots, n$, denote the indegrees of their vertices, respectively. Then $H_1 = H_A \cup H_C =$

$(V, E_A \cup E_C)$, $H_2 = H_B \cup H_C = (V, E_B \cup E_C)$ and by virtue of Statement 2, equations (11) and (12) can be written as follows:

$$\mu_1 2^{\mu_1} = \sum_{i=1}^n 2^{\rho_i^A + \rho_i^B + \rho_i^C}, \quad \mu_2 2^{\mu_2} = \log_2 \left(\sum_{i=1}^n 2^{\rho_i^A + \rho_i^C} + \sum_{i=1}^n 2^{\rho_i^B + \rho_i^C} - \sum_{i=1}^n 2^{\rho_i^C} \right).$$

Let $\rho_i^A + \rho_i^B + \rho_i^C = m_i$ for some fixed $m_i \leq n$, $i = 1, \dots, n$. In this case, the right-hand side in the first equation is a constant value, and, therefore, due to the monotonicity of the function $x2^x$ the emergence coefficient κ will reach its maximum value at the smallest value of the right-hand side of the second equation. To find such a value, it is necessary to solve the problem

$$\varphi = \sum_{i=1}^n 2^{m_i - \rho_i^A} + \sum_{i=1}^n 2^{m_i - \rho_i^B} - \sum_{i=1}^n 2^{m_i - \rho_i^A - \rho_i^B} \xrightarrow{\rho_i^A, \rho_i^B} \min.$$

Let us find the stationary points of the function φ . Having calculated the partial derivatives with respect to ρ_j^A , ρ_j^B , $j = 1, \dots, n$, and equating them to zero, after equivalent transformations we obtain:

$$2^{\rho_j^B} \cdot \sum_{i=1}^n 2^{m_i - \rho_i^A} = \sum_{i=1}^n 2^{m_i - \rho_i^A - \rho_j^B}, \quad 2^{\rho_j^A} \cdot \sum_{i=1}^n 2^{m_i - \rho_i^B} = \sum_{i=1}^n 2^{m_i - \rho_i^A - \rho_j^B}, \quad j = 1, \dots, n. \quad (\text{A.1})$$

Since the right-hand sides are equal, then $2^{-\rho_j^A} \cdot \sum_{i=1}^n 2^{m_i - \rho_i^A} = 2^{-\rho_j^B} \cdot \sum_{i=1}^n 2^{m_i - \rho_i^B}$, $j = 1, \dots, n$. Let us denote $X = \sum_{i=1}^n 2^{m_i - \rho_i^A}$, $Y = \sum_{i=1}^n 2^{m_i - \rho_i^B}$ and rewrite the expression as:

$$2^{m_j - \rho_j^A} \cdot 2^X = 2^{m_j - \rho_j^B} \cdot 2^Y. \quad (\text{A.2})$$

Let us sum the last equality over j , and obtain: $X2^X = Y2^Y$. The function $x2^x$ strictly monotonically increases at $x \geq 0$, therefore the last equation has a unique solution $X = Y$, and therefore, from (14) we obtain that $\rho_j^A = \rho_j^B = \rho_j$, $j = 1, \dots, n$. Let us substitute these conditions into any of the equations (13), and after taking the logarithm and rearranging the terms we obtain:

$\rho_j + \sum_{i=1}^n (2^{m_i - \rho_i} - 2^{m_i - 2\rho_i}) = 0$. At $\rho_i \geq 0$ all terms on the left-hand side are non-negative, therefore the last equality is possible only when all these terms are equal to zero, whence it follows that all $\rho_i = 0$. In the admissible region, this is the only stationary point at which an extremum can be reached. This point lies on the boundary of the region, and the value of the function at it is equal to

$\varphi = \sum_{i=1}^n 2^{m_i}$. However, using the standard technique of studying an extremum using second derivatives, it can be shown that the quadratic form describing the second differential of the function φ at the found point is sign-indefinite, i.e., there is no extremum at this point. This means that the function φ takes an extremal value on the boundary of the region $\rho_i^A, \rho_i^B \geq 0$, $\rho_i^A + \rho_i^B \leq m_i$, i.e., under the condition $\rho_i^A, \rho_i^B \geq 0$, $\rho_i^A + \rho_i^B = m_i$, $i = 1, \dots, n$.

To find a stationary point on the boundary, we substitute $\rho_i^B = m_i - \rho_i^A$ into the expression for the function f : $f = \sum_{i=1}^n 2^{m_i - \rho_i^A} + \sum_{i=1}^n 2^{\rho_i^A} - 2^n$. Having calculated the derivatives with respect to ρ_j^A and equated them to zero, after transformations we obtain a system of equations for finding stationary points:

$$\sum_{i=1}^n 2^{m_i - \rho_i^A} \cdot 2^{m_j - \rho_j^A} = \sum_{i=1}^n 2^{\rho_i^A} \cdot 2^{\rho_j^A}, \quad j = 1, \dots, n. \quad (\text{A.3})$$

We sum up all the equations with respect to j and denote $X = \sum_{i=1}^n 2^{m_i - \rho_i^A}$ and $Y = \sum_{i=1}^n 2^{\rho_i^A}$, as a result we obtain $X2^X = Y2^Y$. This equality is possible only for $X = Y$. Therefore, from (15) we

obtain $2^{m_j - \rho_j^A} = 2^{\rho_j^A}$, whence $\rho_j^A = \frac{m_j}{2} = \rho_j^B$, $j = 1, \dots, n$. Routine research using second derivatives shows that at this point the minimum of the function $\varphi_{\min} = 2^{1 + \sum_{i=1}^n 2^{\frac{m_i}{2}}} - 2^n$ is indeed reached. Thus, to find the greatest value of the emergence coefficient κ under the condition $\rho_i^A + \rho_i^B + \rho_i^C = m_i$ we obtain the equations:

$$\mu_1 2^{\mu_1} = \sum_{i=1}^n 2^{m_i}, \quad (\text{A.4})$$

$$\mu_2 2^{\mu_2} = \log_2 \left(2 \cdot 2^{\sum_{i=1}^n 2^{\frac{m_i}{2}}} - 2^n \right). \quad (\text{A.5})$$

Let us estimate the right-hand side of (17) from below. Since $2^{\sum_{i=1}^n 2^{\frac{m_i}{2}}} - 2^n \geq 0$ for any m_i , $i = 1, \dots, n$, then

$$\log_2 \left(2 \cdot 2^{\sum_{i=1}^n 2^{\frac{m_i}{2}}} - 2^n \right) \geq \log_2 2^{\sum_{i=1}^n 2^{\frac{m_i}{2}}} = \sum_{i=1}^n 2^{\frac{m_i}{2}} > \left(\sum_{i=1}^n 2^{m_i} \right)^{\frac{1}{2}}.$$

Along with equation (17), we consider equation $\bar{\mu}_2 2^{\bar{\mu}_2} = \left(\sum_{i=1}^n 2^{m_i} \right)^{\frac{1}{2}}$. Due to the monotonicity of the function x^{2^x} we conclude that $\bar{\mu}_2 < \mu_2$. Consequently, $\kappa = \frac{\mu_1}{\mu_2} < \frac{\mu_1}{\bar{\mu}_2} = \bar{\kappa}$, so $\bar{\kappa}$ can be considered as an upper bound for κ .

Now we divide equation (16) by $(\bar{\mu}_2 2^{\bar{\mu}_2})^2$: $\frac{\mu_1 2^{\mu_1}}{\bar{\mu}_2^2 2^{2\bar{\mu}_2}} = \frac{\bar{\kappa}}{\bar{\mu}_2} 2^{\bar{\mu}_2(\bar{\kappa}-2)} = 1$. We consider the last equation as an implicitly defined function $\bar{\kappa}$, depending on $\bar{\mu}_2$, i.e., $F(\bar{\mu}_2, \bar{\kappa}) = \frac{\bar{\kappa}}{\bar{\mu}_2} 2^{\bar{\mu}_2(\bar{\kappa}-2)} - 1 = 0$. We find the extremum of the implicit function $\bar{\kappa}$ of $\bar{\mu}_2$. In order for $\bar{\kappa}'_{\bar{\mu}_2} = 0$, the equality $F'_{\bar{\mu}_2} = -\frac{\bar{\kappa}}{\bar{\mu}_2^2} 2^{\bar{\mu}_2(\bar{\kappa}-2)} + \frac{\bar{\kappa}}{\bar{\mu}_2} 2^{\bar{\mu}_2(\bar{\kappa}-2)} (\bar{\kappa} - 2) \ln 2 = 0$ must be satisfied. Solving the equations $F'_{\bar{\mu}_2} = 0$ and $F = 0$, together, we find two solutions to these equations: $\bar{\kappa}_{1,2} = 1 \pm \sqrt{1 + \frac{1}{e \ln 2}}$. Since the emergence coefficient is obviously a positive value, we finally obtain $\kappa < \bar{\kappa} = 1 + \sqrt{1 + \frac{1}{e \ln 2}} \approx 2.2372$. Using standard methods, we can show that the found value is the only maximum of the function under study, which proves the upper bound for κ .

Theorem 1 is proven.

REFERENCES

1. Damper, R.I., Editorial for the Special Issue on "Emergent Properties of Complex Systems": Emergence and levels of abstraction, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 811–818.
<https://doi.org/10.1080/002077200406543>
2. Crick, F., *The Astonishing Hypothesis. The Scientific Search for the Soul*, London: Simon and Schuster, 1994.
3. Goldstein, J.A., Emergence and Radical Novelty: From Theory to Methods, *Handbook of Research Methods in Complexity Science*. Edward Elgar Publishing, 2018, Chapter 23, pp. 507–524.
4. Chalmers, D.J., Strong and Weak Emergence, in: *The Re-Emergence of Emergence: The Emergentist Hypothesis from Science to Religion*, Oxford: Oxford University Press, 2006, pp. 244–254.
5. Bedau, M.A., Weak Emergence, *JSTOR. Philosophical Perspectives: Mind, Causation, and World*, 1997, vol. 11, pp. 375–399.

6. Bar-Yam, Y., A Mathematical Theory of Strong Emergence Using Multiscale Variety, *Complexity*, 2004, vol. 9, no. 6, pp. 15–24.
7. Johnson, J.J. IV, Tolk, A., and Sousa-Poza, A., A Theory of Emergence and Entropy in Systems of Systems, *Procedia Comput. Sci.*, 2013, no. 20, pp. 283–289.
8. Stepney, S., Polack, F.A.C., and Turner, H.R., Engineering Emergence, *11th International Conference on Engineering of Complex Computer Systems (ICECCS 2006)*, Stanford, USA, 2006.
9. Fromm, J., On Engineering and Emergence, *arXiv preprint nlin/0601002*, 2006.
10. Alexander, J.R., Jr. and Challef, S., Control: An Emergent Property of Biological Neurons, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 895–909.
11. Chen, Y. and Reggia, J.A., The Temporal Correlation Hypothesis for Self-Organizing Feature Maps, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 911–921.
12. Watters, P.A., Time-Invariant Long-Range Correlations in Electroencephalogram Dynamics, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 819–825.
13. Channon, A.D. and Damper, R.I., Towards the Evolutionary Emergence of Increasingly Complex Advantageous Behaviours, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 843–860.
14. Cucker, F. and Smale, S., Emergent Behavior in Flocks, *IEEE Trans. Autom. Control*, 2007, vol. 52, no. 5, pp. 852–862.
15. Cucker, F. and Smale, S., On the Mathematics of Emergence, *Jpn. J. Math.*, 2007, vol. 2, pp. 197–227.
16. Ke, J., Minett, J., Au, C.-P., and Wang, W.S.-Y., Self-Organization and Selection in the Emergence of Vocabulary, *Complexity*, 2002, vol. 7, no. 3, pp. 41–54.
17. Benjamin, D.P., On the Emergence of Intelligent Global Behaviours from Simple Local Actions, *Int. J. Syst. Sci.*, 2000, vol. 31, no. 7, pp. 861–872. <https://doi.org/10.1080/002077200406589>
18. Surmin, Yu.P., *Systems theory and systems analysis: Textbook*, Kiev: MAUP, 2003.
19. Marcus, M. and Mink, H., *A Survey of Matrix Theory and Matrix Inequalities*, Boston: Allys and Bacon, 1964.
20. Cavanagh, J., *Sequential Logic. Analysis and Synthesis*, CRC Press, Taylor&Francis Group, 2007.
21. Malyugin, V.D., Implementation of Boolean functions by arithmetic polynomials, *Avtomat. i Telemekh.*, 1982, no. 4, pp. 84–93.
22. Rubio, A.J., Algebraic Representation of Boolean Functions, *Int. J. Electron.*, 1984, vol. 56, no. 5, pp. 735–739. <https://doi.org/10.1080/00207218408938868>
23. Jain, J., Arithmetic Transform of Boolean Functions, *Representations of Discrete Functions*, Boston: Springer, 1996, pp. 133–161. https://doi.org/10.1007/978-1-4613-1385-4_6

This paper was recommended for publication by D.V. Vinogradov, a member of the Editorial Board